AD-A249 727

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 14 Oct 91	3. REPORT TYPE AND DATES COVERED THESIS/~~DISSERTATION~~
4. TITLE AND SUBTITLE An Information Framework for Facility Programming		5. FUNDING NUMBERS
6. AUTHOR(S) Gregory M. Perkinson, Capt		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AFIT Student Attending: Pennsylvania State University		8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/CI/CIA-91-101
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFIT/CI Wright-Patterson AFB OH 45433-6583		10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES		
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release IAW 190-1 Distributed Unlimited ERNEST A. HAYGOOD, Captain, USAF Executive Officer		12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words)		
14. SUBJECT TERMS		15. NUMBER OF PAGES 133
		16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT
20. LIMITATION OF ABSTRACT		

The Pennsylvania State University

The Graduate School

College of Engineering

AN INFORMATION FRAMEWORK FOR FACILITY PROGRAMMING

A Thesis in

Architectural Engineering

by

Gregory M. Perkinson

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 1991

We approve the thesis of Gregory M. Perkinson.

Date of Signature

Francois Grobler — 9/16/1991

Francois Grobler
Assistant Professor of
Architectural Engineering
Thesis Advisor

Victor E Sanvido — 9/16/91

Victor E. Sanvido
Associate Professor of
Architectural Engineering

C. Herbert Wheeler — 9-16-91

C. Herbert Wheeler
Emeritus Professor of
Architectural Engineering

Paul A. Seaburg — 9/16/91

Paul A. Seaburg
Professor of Architectural Engineering
Head of Department of
Architectural Engineering

From: Capt Perkinson (AFIT/CI at Penn State) 14 Oct 91

Subject: Thesis Submission

To: AFIT/CIRD

1. My MS thesis was approved by the university late last week. The information you require is enclosed:

   a. Cover page with name, rank, etc.

   b. Two copies of the abstract

   c. A copy of the bibliography, noting primary and secondary sources

   d. Last, but not least, an unbound copy of the approved thesis

2. Here is a listing of government agencies that might be interested in this research:

   USAF Base Civil Engineering
   The Air Force Academy (Department of Civil Engineering)
   AFIT (School of Civil Engineering)
   The US Army Civil Engineering Research Laboratory (CERL)

If you have any questions, please don't hesitate to contact me at (comm) (814) 865-3369 (office), or (814) 238-5895 (home).

GREGORY M. PERKINSON, Capt, USAF
AFIT Student

4 Atch:
1. Cover page
2. Abstract
3. Bibliography
4. Thesis

The Pennsylvania State University
The Graduate School
College of Engineering

AN INFORMATION FRAMEWORK FOR FACILITY PROGRAMMING

A Thesis in
Architectural Engineering

by
Gregory M. Perkinson

Captain, United States Air Force

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

1991

ABSTRACT

This thesis presents an "open" information framework to store, manage, and retrieve facility programming information. The Facility Programming Product Model (FPPM) represents information that is normally contained in the facility program in a form that allows members of the facility team (owner, planner, designer, constructor and operator) to access relevant information.

The FPPM is a systematized approach to creating, organizing, and presenting facility programming information. It allows the owncr to critique design, construction, and operation based on the programmed building functions. The FPPM defines a coding system, as well as an information display for 6 types of programming information. The classification system has two parts: 1. the address code, which act as an "address" to categorize information; 2. the utility code, which represents the priority information has relative to the project.

Information categories in the FPPM were derived from current literature, then refined through a review of facility programs for 15 existing projects. The model was reviewed and critiqued by industry experts. Then, the model was applied to 8 case studies through interviews with project level owner's representatives. The purpose was twofold: verify **completeness** (that necessary information was included) and identify **criticality** (what information was essentiai and why). The lessons learned from the case studies were then structured as guidelines for the owner's representative..

The guide was designed as a checklist-like set of rules that lead the owner's representative through the development of the program. It can also be used to analyze the process or product of subsequent work to ensure it meets the original goals/objectives established when developing the model.

ABSTRACT

This thesis presents an "open" information framework to store, manage, and retrieve facility programming information. The Facility Programming Product Model (FPPM) represents information that is normally contained in the facility program in a form that allows members of the facility team (owner, planner, designer, constructor and operator) to access relevant information.

The FPPM is a systematized approach to creating, organizing, and presenting facility programming information. It allows the owner's representative to review the building requirements (the program) for completeness by establishing a structure designed to access programming criteria at varying levels of abstraction, during any phase of the building life cycle. It also allows the owner to critique design, construction, and operation based on the programmed building functions. The FPPM defines a coding system as well as an information display for six types of programming information. The classification system has two parts: the address code, which acts as an "address" to categorize information; and the utility code, which represents the priority information has relative to the project. The coding elements follow.

The address coding scheme is comprised of "level," "general categories," "graphic link," and "system" codes. Level defines the level of detail demonstrated by the program information, and can be initial/schematic or detailed. The general categories of programming information include Preprogram, Function, Economy, Schedule, Form, and Social Issues. Graphic link is a reference to a graphic image. The system code defines the discipline involved in a particular aspect of the program, i.e., civil,

architectural, electrical, etc. The utility coding scheme consists of "priority" and "value." Priority is categorized as one of four levels: 1. Mission Essential; 2. Safety/Health; 3. Valid Requirement; or 4. Nice-to-have. The code value is a relative means of comparing different categories using a common basis, usually cost.

Information categories were derived from current literature, then refined through a review of facility programs for 15 existing projects. The model was first reviewed and critiqued by industry experts, and then applied to a case study of public sector projects through interviews with project level owner's representatives. The purpose was to verify **completeness** and identify **criticality** of information. The lessons learned from the case study were structured as guidelines for the owner's representative in the form of a facility programming guide.

The guide was designed as a checklist-like set of procedures for gathering information and a suggested format for presenting that information. The guidelines could lead the owner's representative through the development of the program. After the program is developed, recommended criteria were presented showing key decision points when design, construction, and/or operations information could be evaluated against the program to ensure that the original goals/objectives established were met.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Special thanks are owed to Dr. Francois Grobler, who provided countless hours as my principal advisor, providing focus and fresh ideas regarding academic research.

Dr. Victor E. Sanvido provided the full services of the Computer Integrated Construction Laboratory, comprising resources, personal contacts, and consistency among similar research topics in the laboratory. His time and energy clarified issues related to the phases of a building's construction life cycle and deserved thanks and recognition.

I would also like to acknowledge the contribution of Professor C. Herbert Wheeler, whose expertise as a programmer, designer and educator is unparalleled in the industry. He dedicated numerous hours as my facility programming mentor and his efforts are truly appreciated. Two other experts in the profession contributed to this work by reviewing the model and offering their insight. They are William P. Ross and Raymond Bowers. Their contributions are greatly appreciated.

Finally, I thank the Air Force Institute of Technology for providing the funding and support which enabled me to participate in the graduate program at Penn State.

BIBLIOGRAPHY

Primary Sources

American Institute of Architects (AIA), DOCUMENT B141, "Owner-Architect Agreement," 14th ed., Washington, DC: AIA Press, 1987, pg. 5.

Evans, Benjamin H. and Wheeler, C. Herbert, Emerging Techniques - 2, Washington, DC: AIA Press, 1969.

Beckett, James P., "An Information Framework for Facility Operators," Computer Integrated Construction, Technical Report No. 23, Penn State, 1991.

Khayyal, Sari A, "Towards a Master Builder Information Framework for Project Developers," Computer Integrated Construction, Technical Report No. 16, Penn State, 1990.

Palmer, Mickey A., The Architect's Guide to Facility Programming, Washington DC: AIA Press and New York: Architectural Record Books, 1981.

Peña, William, et al., Problem Seeking- An Architectural Programming Primer, 3rd ed., Washington, DC: AIA Press, 1987.

Perkinson, Gregory M., "A Summary of Program Evaluations," Working Paper, Department of Architectural Engineering, Penn State, May, 1991.

Sanoff, Henry, Methods of Architectural Programming, Stroudsburg: Dowden, Hutchinson, and Ross Inc. (definition by the community development series editor, Richard Dopler), 1977.

Sanvido, Victor E., "An Integrated Building Process Model," Computer Integrated Construction, Penn State, 1990a.

Sanvido, Victor E., "Critical Project Success Factors (CPSF) for Buildings," Computer Integrated Construction, Technical Report No. 22, Penn State, 1990b.

White, Edward T., Introduction to Architectural Programming, University of Arizona, Tuscon: Architectural Media Ltd., 1972.

White, Edward T., "Learning Decision-making for the Building Process," Florida A & M University, undated (b).

White, Edward T., "The Value of Post Occupancy Evaluation to the Architect in Government," Florida A & M University, undated (c).

Secondary Sources

American Institute of Architects (AIA), Handbook of Professional Practice, vol. 1-3, Washington, DC: AIA Press, 1984.

Boyd, M.D. and James, E.M., "A conceptual Framework for Post-Occupancy Evaluation and an Analysis of Eleven Studies," Architectural Science Review, vol. 31, 1988, pp. 133-143.

Daish, John et al., "Post Occupancy Evaluation in New Zealand," Design Studies, vol. 3, no. 2, April 1982.

Davis, Gerald, "A Process for Adapting Existing Buildings for New Office Uses," taken from Facility Programming, Stroudsburg: Dowden, Hutchinson, and Ross Inc., 1978.

Gordon, Douglas E. and Stubbs, M. Stephanie, "Programming," Architecture, May 1988, pp. 203-210.

Peña, William M. and Focke, John W., "Performance Requirements of Buildings and the Whole Problem," National Bureau of Standards, Special Publication, vol. 361, no.1, 1972, pp. 43-55.

Peña, William M., "Programming Management and Techniques," Conference notes from a national colloquy: Emerging Techniques of Practice Management, Penn State, 1969.

Preiser, Wolfgang F. E., Facility Programming, Stroudsburg: Dowden, Hutchinson, and Ross Inc., 1978.

Preiser, Wolfgang F. E., Programming the Built Environment, New York: Van Nostrand Reinhold Company, 1985.

Preiser, Wolfgang F. E., "The habitability framework: a conceptual approach towards linking human behavior and physical environment," Design Studies, Butterworth and Co., vol. 4, no. 2, April, 1983.

Royal Architectural Institute of Canada, Canadian Handbook of Practice for Architects, vol. 1, "Requirements and Predesign," March, 1977.

Royal Institute of British Architects, "The Project (Planning)-2, The Techniques and Methods," vol. 4, part 3 (3.500), July 1965.

Sanvido, Victor E., "Towards a process based information architecture for construction," Civil Engineering Systems, E. and F. N. Spoon Ltd., 1990c.

Simon, H. A., The New Science of Management Decision Systems, New York: Harper and Row, 1960.

White, Edward T., "Building Evaluation in Professional Practice," Florida A & M University, undated (a).

White, Edward T., Space Adjacency Analysis, Tuscon: Architectural Media Ltd., 1986.

White, Edward T., "Programming Post Occupancy Evaluation and the Financial Success of the Architect," Tuscon: Architectural Media Limited, 1988.

White, Edward T., "Post Occupancy Evaluation from the Client's Perspective," Tuscon: Architectural Media Limited, 1989.

Chapter 1

INTRODUCTION

1.1. INTRODUCTION

The process of programming varies depending on the programmer's view and his intent. Historically, the owner was responsible for the facility program. Presently, programs may be developed by the owner, a programming/planning firm, architectural firms with programming departments, or traditional architectural firms. The facility program was developed as a predesign service. It was subsequently used during schematic design and design development as a basis for design decisions.

This thesis studied the information needs associated with the facility program and presented a framework for that information. It explored how programs can be used during the life cycle of construction to objectively evaluate contractor and facility performance criteria (set in the program). The information framework was based on the literature, sample facility programs, and industry experts' review. Case study projects were used to develop the base material for a facility programming guide, which is presented.

1.2. BACKGROUND

Programming began as a listing of the owner's physical (functional) and economic criteria. It has evolved to include social, psychological and aesthetic factors (Palmer, 1981). Facility programs contain general and specific information regarding the building's requirements.

Informally, programming of facilities has been done as long as architecture has existed. According to Palmer (1981), formal programming (as we know it today) evolved around the time Peña wrote his first article on programming in 1959. However, researchers don't agree on any "best" programming technique. Significant contributions to programming have been made by Evans, Wheeler, Peña, Focke, Sanoff, Preiser, White, Davis, and others.

The program is developed during the planning phase of the life cycle of a facility (Sanvido, 1990a). The architect uses the program to develop design solutions to the stated problem. The link between programming and design is strong. This relationship should not be underestimated. Unfortunately, the traditional use for the program is primarily during design development. The program is rarely used to evaluate contractor performance. It also has infrequent use as a tool for post occupancy evaluation (Boyd and James, 1988).

This author believes that the program can be better used during the design phase to evaluate design alternatives; and to check whether the solutions presented satisfy the original design intent stated in the program. The potential applications for the program during the facility's contruction and operation have not been realized.

The interpretation of programming terminology varies. Terms used in this thesis are defined in Appendix A. Programming definitions vary, depending on what architect or design firm is asked. Examples are presented in Appendix B. The American Institute of Architects (AIA) defines the owner's and Architect's responsibilities in their standard contract, AIA Document B141 (1987). They indicate in Article 4:

> The owner shall provide full information regarding requirements for the Project, including a program which shall set forth the Owner's objectives, schedule, constraints and criteria, including space requirements and relationships, flexibility, expandability, special equipment, systems and the site requirements.

This author provides a working definition of programming later in section 1.5.1.

1.2.1. Life Cycle of Construction

According to Sanvido (1990a) the life cycle of providing a facility encompasses the following processes: manage, plan (the program is a product of this phase), design, construct, and operate. This context of the construction life cycle, described by the Integrated Building Process Model (IBPM), was the basis for referring to the life cycle of providing a facility used in this thesis.

1.2.2. Product Models

Product models were initially developed for the Architecture, Engineering, and Construction (AEC) industry in the mid 1980s. These models attempted to represent the physical characteristics of a given product, which were normally the output of some process and focussed on identifying the physical characteristics of a building.

In similar terms, a product model for programming should represent the physical characteristics of the program, because the program is the product (output) of the programming process. Product models related to the AEC industry are identified in Chapter 2.

1.3. PROBLEM STATEMENT

The program document doesn't "survive" beyond the design phase. Consequently, the facility team may lose their focus on the owner's original (or modified) planning, designing, constructing, or operating intentions. Additionally, the program is not typically used as a means to provide information to the facility team which could help resolve conflicts that occur in subsequent phases of construction.

1.4. SIGNIFICANCE

The information in the program can be utilized throughout the life cycle of construction if the information elements meet the needs of the facility team.

The following examples present the author's view of what these uses are:

- The program document can provide a framework for tracking the facility team's goals ("staying on course").
- The program can also be applied as a cross check during the phases of design, construction, and operation; but it should be flexible enough to accommodate the facility team's needs.
- It should consider the "hard and soft" general programming categories of information: function, economy, schedule, social issues, and form.
- Examples of critical information the program should consider are future expansion, equipment operation, equipment maintenance, and tracking facility repairs.
- The level of involvement should be primarily within the "heart" of the facility team (project level owner's representative).

Based on the increased complexity, quantity and variety of information in the AEC industry, programming requires a systematized process of developing and managing data/information (Palmer, 1981). The industry is interested in techniques that will add to the facility team's experience and optimize the knowledge they have about facility designability, constructability, and operability (Critical Project Success Factor Study, Sanvido, 1990b). The owner, programmer, designer, builder, and operator will enhance the facility team's collective experience by using the program. Therefore, keeping the program document "alive" during the life cycle could have significant impact when considering future facility maintenance and operations, and subsequent facility renovations, modifications, or expansion.

1.5. OBJECTIVES

The goal of this research is to identify and confirm the programmer's information needs, then to test the feasibilty of using that information during the entire life cycle of a facility. Programming information is first defined through the development of a Facility Programming Product Model (FPPM). The following are the objectives of this thesis:

- Define programming.
- Develop the FPPM--a conceptual framework that reflects an organized approach to carrying program information at various levels of detail throughout the construction life cycle and is suited to public sector work .
- Test the FPPM using a public sector case study.
- Develop a programming guide, based on the "lessons learned" from the case studies. The guide establishes a comprehensive way to accumulate and classify information needed by the facility team during the life of a project. It will be presented in the form of a flexible set of guidelines that lend themselves to creating a "facility file."

1.5.1. Defining Programming

Programming should be defined, in order to create a basis for understanding the author's use of the term. This definition of programming is a working definition for the study. A formal definition, developed as a result of the study, is presented in Chapter 6.

Facility programming is the process of analyzing the owner's desires, needs, goals and objectives in order to define essential facility requirements; presenting those criteria to the designer; then establishing and maintaining a framework which carries that information throughout the life cycle of construction.

1.5.2. Modeling the Facility Program

The FPPM should represent an "open" information framework that members of the facility team can utilize to satisfy their individual goals and the construction project's goals. The model should show a product that has the capacity to carry information forward to each stage of the life cycle. It should also contain information the owner's representative may need to effectively communicate with other members of the facility team, resulting in satisfying the owner's facility goals and requirements. Those needs include the ability to gather, store, retrieve and/or modify the general or specific facility requirements.

Information related to specific categories (at various levels), would be applied to a given project as the project's information needs dictate. For example, if master planning (pre-programming) information is not available, or needed, by the facility team, that portion of the framework would not be utilized. Only information needed by the team would be applied to the framework.

The FPPM should focus on the compilation and evaluation of programming information. The evaluation should be supported by a priority coding scheme and the inclusion of a specific "checkpoint" for placing a value on the programming information gathered. This checkpoint (the "value" cell) is explained in Chapter 3.

1.5.3. Testing the FPPM

The FPPM should be tested using a case study of public sector projects in a major university campus setting. Interviews should be conducted with each project's Owner's Representative (OR) to determine the essential information elements needed in the facility program. The case study data can then be compared to the FPPM to determine how complete each program is, and what information is critical to the OR.

1.5.4. Programming Guide

The lessons learned from the case study should be used in conjunction with the information framework in the FPPM to develop a programming guide. The guide can be written for the OR view point. It is intended to be used by planners, programmers, project managers or the facility coordinator.

1.6. SCOPE

The model was developed using a generic framework that can be developed further for manual or automated applications. This research focussed on the process for developing the information categories and the format of the FPPM. It also presented a format and guidelines for using that information.

This study was limited to a major university's facility management program. The case study utilized projects on the main campus. These projects were funded be the public. The case study tested the viewpoint of project level owner's representatives. A pair of projects were taken from each of the following phases of the construction life cycle:

- planning/conceptual design
- design development
- construction
- post occupancy

1.7. METHODOLOGY

This section identifies how the research was conducted and describes the development of the FPPM. The methodology for testing the FPPM and developing the guide is also presented.

1.7.1. Investigation and Model Development

An initial review of the literature provided the basis for developing the conceptual model and a preliminary definition of programming. The FPPM was developed based on information elements derived from the literature, and the author's personal experience. The model was refined after a study of 15 facility programs to identify an appropriate format and content for a program (working paper by Perkinson, 1991). Subsequently, the model was reviewed and critiqued by a team of three industry experts to validate its framework and contents. Feedback from this review was used to modify the FPPM before the case study evaluation process began.

1.7.2. Case Study

The case study consisted of four pairs of projects in various phases of the construction life cycle. The case study was conducted in three phases. First, project files were reviewed to familiarize the author with the history and scope of work. This entailed a detailed review of documentation available in the project file. The next phase involved interviewing the ORs of the various projects. In the last phase, the interview data was analyzed to determine what information in the program was critical to the OR and what aspects of the program could be useful in the construction and operations phases of the facility.

1.7.3. Programming Guide

A programming guide was developed based on the lessons learned from the industry representatives' reviews of the FPPM and the case study evaluation. The guide was based on the same "open" (flexible) framework established by the FPPM and acted as a foundation, or road map for describing what information should be in the program.

1.8. ORGANIZATION OF THESIS

Chapter 2 identifies the literature which contributed to the development of the information framework for facility programming. It identifies the programming process and briefly discusses product modeling literature, then summarizes the life-cycle of construction viewpoint used in this thesis.

Chapter 3 discusses the development, structure of the FPPM, and rules for using it. In Chapter 4, the case study results and implications of the case study analysis are discussed.

Chapter 5 presents the programming guide, which was developed for the owner's representative of public sector work. The guide is based on lessons learned through refinements to the FPPM and results of the case study. Chapter 6 concludes the thesis with a summary, observations about the value of this research, and suggested areas of future research.

Chapter 2

LITERATURE REVIEW

2.1. INTRODUCTION

This chapter outlines the key writings related to process and product orientations in programming. Facility programming guidelines from the architectural profession are summarized. Then, product modeling efforts relevant to this thesis are presented. The reader is introduced to a view of the life cycle of construction. Lastly, the final discussion highlights what is lacking in the industry, and how this thesis fills that gap.

2.2. PROGRAMMING LITERATURE

Programming literature focuses primarily on the process of programming. The product of programming, what is produced in the process, is rarely emphasized. Important writings in the field of programming from six authors, and their significance to this research, are presented in this section. Programming guidelines from three architectural societies are also discussed.

2.2.1. Evans and Wheeler:

Programming techniques vary across the industry, and were categorized by Evans and Wheeler (1969) as fitting into the following six different groups of techniques:

- Standard procedures
- Data banking techniques
- Planning techniques
- Investigative techniques
- Analytical techniques
- Presentation techniques.

As the first authors to document various programming techniques, Evans and Wheeler (1969) discovered four problem areas common to architects and planners:

1. Communication (getting at the client or user's real desires)
2. Problem definition and hierarchy (identifying the real problem)
3. Fact collection
4. Fees and services (there is no established standard).

The American Institute of Architects (AIA) standard contracts have corrected the fourth area of concern, by defining the architect's and owner's roles. The standard contract B141 states that the owner shall provide the program. However, many other options are available to an owner (hire a programming firm, the design firm, etc.).

Evans and Wheeler's work was significant to this research because it established the first review of the "state of the art" in programming. Even though the research was done in the mid 1960s, the types of programming processes and products discovered then are still prevalent today. The problem areas still exist today as well.

2.2.2. Peña

Peña (1987) approached programming as a "problem seeking" method. He identified a five-step method for researching the facility requirements, based on four considerations (function, form, economy and time). He distinctly separated programming (problem seeking) from design (problem solving).

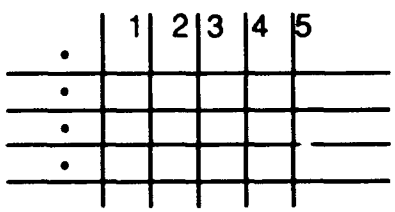
Peña defined a method of applying the process and considerations of programming in an information framework. A schematic view of the framework is shown as Figure 2.1 (with the approaches of the two authors discussed next). This matrix forms a checklist the programmer can use to evaluate his process and ensure he addresses the issues identified by the matrix. Peña emphasized building a team (with the designer and client) and how critical communication was to the process.

Peña stressed the importance of the fifth step in the process: stating the problem. He contended the problem statement was the product of programming and the link with design.

Pena, CRSS "Problem Seeking"

Five-step Process:
1 Establish **Goals**
2 Collect and Analyze **Facts**
3 Uncover and test **Concepts**
4 Determine **Needs**
5 State the **Problem**

Four Considerations:
- **Function** (people, activities, relationships)
- **Form** (site environment, quality)
- **Economy** (initial budget, operating costs, life cycle costs)
- **Time** (past, present, future)



Information Index

White, Florida A & M Univ.
Five-step Process:
1. Gather
2 Analyze
3 Evaluate
4 Organize
5 Present

INFORMATION

Preiser, "Habitability Framework"

Seven-step Process:
1. Obtain Org **goals and objectives**
2. Translate objectives into **functions**
3. **Break down** functions into:
   - activities
   - programs
4. Formulate **performance criteria** and environmental criteria.
5. Devise **adjacency requirements** and **schedule** (prioritize time/space utilization)
6. **Designate spaces** (gross area estimate)
7. Present **Options** (tied to time and cost)

Figure 2.1: Programming Process--A Sample of Various Techniques

Peña also described programming as a two-step process. The first was collecting and analyzing information at the schematic level. This schematic information "feeds" the schematic design solutions. The second level of detail is the refinement and development of the initial information. This detailed programming information feeds the more detailed design solutions.

Peña was perhaps the most respected author in the field of architectural programming today. His Problem Seeking- An Architectural Programming Primer brought the *process* of programming into widespread awareness. His methodology and rationale for developing and presenting a program were clearly presented. However, he doesn't offer any clear guidance on the programming *product.* In this thesis, process and product are discussed together in some cases, but the focus is on the product.

Peña's four considerations in programming (form, function, economy, and time) clearly categorized the vast majority of information needed in a program. Other general categories were subsequently identified as being relevant to a programming product model (see Chapter 3) but Peña's form, function, economy, and time still form a valid foundation for programming.

The concept of differentiating between the different levels of programming information was based on Peña's writings also. For example, schematic programming information reflected in the initial process of gathering information about the functional requirements might reflect the need to provide heating and cooling for a space. This schematic programming information would then alert the designer that environmental control issues will need to be resolved and that a system will have to be developed to solve the problem. As more information is gathered, the schematic programming information is developed more fully as the detailed program requirements (i.e., what the ideal temperature range is for the occupants, what indoor air quality provisions need to be addressed, etc.).

The detailed design solutions would utilize the specific programming information. It is important to note that the process of identifying the problems (programming) and developing solutions (design) may require many iterations. Moreover, when changes in scope or additional information affects either programming or design information, the process of analyzing and synthesizing that information often goes back to the starting point. The schematic and detailed levels of programming were reflected in the FPPM and the link between programming and design became an area of interest when conducting interviews during the case study.

2.2.3. White

White (1972) developed an introductory tool for programmers. His writings discussed the role of programming at that time. He provided basic programming theory and broke it into three areas: the **value** of programming; the **operations** that go into program development; and the **relationships** between programming schematic design and design development. White's view of the programming process is outlined on figure 2.1 for comparative reasons.

White (1986) stressed the importance of using graphic tools to reinforce written concepts (prose) when presenting programming information. He specifically addressed the use of matrices when analyzing space adjacency issues.

The diversity of White's published work about programming (much of it in support of the academic curricula on programming at Florida A & M University) provided an important foundation, or stepping stone, in the path towards understanding programming (both the process and product). White's suggested method of analyzing programs (for content, style, etc.) was adopted in a study by this author to determine if there were common aspects among various programming documents. The results of the study were documented in a working paper by Perkinson (1991).

2.2.4. Palmer

Palmer (1981) presented a comprehensive analysis of programming, based on his review of various techniques (processes) of programming and various formats (products); then, with an edited series of sample programs (designed to provide an overview of various techniques).

Throughout his work, Palmer stressed that programming was essentially a systematized way to handle complex information. Overall, the book discussed the advances made in the 1970s (since the AIA published Emerging Techniques-2). Palmer noted that the program determined the scope and function of a facility, as well as assisting the owner to determine the feasibility.

Palmer viewed programming as an "information processing system." It was a method to accumulate data, then organize, translate and communicate the information.

Palmer explained the evolution of programming masterfully, noting the traditional role (as a list of client requirements) and then discussing the modern role(s). He noted that the scope of programming objectives has extended to: investigating and developing information, analyzing owner and user needs, and evaluating design after construction and occupancy.

In The Architect's Guide to Facility Programming, Palmer presented an excellent overview of programming process, product, and then presented a series of cases studies of sample programs. However, he does not address the concept of using the program (product) as a means to evaluate work in subsequent phases of the life-cycle of the construction process.

Palmer presented the clearest view of why programming has evolved to its present state--based on complex information requirements. He also put programming in a useful generic perspective, as a systematized approach to gathering, storing, and retrieving information. This perspective is the essence of how the FPPM can be used as a product to support programming.

2.2.5. Preiser

Preiser (1978) served as the editor for a compilation of articles about facility programming. He also provided an introductory chapter about the background behind current programming concepts and the evolution leading to current programming practices. His presentation of other authors' concepts was divided into three main areas: "Facility Programming" (a how-to guide from five firms in practice); "Programming for Architecture and

Design" (discussed predesign issues in project development); and Research for Facility Programming" (described various research methods).

Preiser was the first to collect and publish diverse views about programming. His focus on human behavior and value systems contributed to the development of the general category of programming information in the FPPM entitled "social issues".

2.2.6. Sanoff

Sanoff (1977) focused on the use of decision making tools to assist the programming process. First , he discussed "preconditions to programming" where the emphasis was on techniques to organize the programming firm's resources. This section is followed by "Information retrieval methods" which discussed how to identify and explore the design problem, search for ideas, classify information, and generate (and evaluate) alternatives. The information retrieval methods were categorized as follows:

- collective decision methods
- comparison methods
- rating methods
- visual preference methods
- descriptive and evaluation methods
- design methods.

Lastly, Sanoff discussed "methods of transforming design information" by explaining the link between programming and design. He presented six tools or "design models", which were based on client input/communication. Then, five case studies were used to reinforce programming concepts.

Sanoff was interested in the facility user's perspective. He found that a relatively small number of owner/operator organizations develop facility data systems to assist their operation of the finished facility. Therefore, the program could be the basis for these systems. An operator's actions might include: documenting as-built information; developing data showing "systems" information; showing facility restrictions (and attributes) relating to operations; and developing operations procedures (manuals/checklists). Beckett (1991) identified the importance of developing an information framework for the facility operator; and his model served as the foundation for the frame in the FPPM.

2.2.7. Programming Guidelines

Three different professional architectural societies are compared in this section. These organizations offer guidelines to the programming and design professional in their respective countries. The extent to which each organization discusses programming follows.

2.2.7.1. The American Institute of Architects (AIA)

The AIA does not offer much direction regarding either the process or product for programming in it's standard contract documents. The AIA description of programming is found in Appendix B: "Programming Definitions". The A Press sponsored guidelines for programming in Peña's Problem Seeking (all three editions) and Palmer's Architect's Guide to Facility Programming (1981); but neither of these guides was presented to the practicing architect by the AIA.

It is unfortunate that the AIA's Handbook of Professional Practice (1984) did not address programming in the degree of detail as the Canadian and British architectural societies do. As a result, American architects and planners are left with limited programming resources. They have to rely on their background in higher education and any personal skills they've developed through continuing education.

2.2.7.2. The Royal Architectural Institute of Canada

The Royal Architectural Institute of Canada (RAIC, 1977) does discuss the programming process and expectations (results) in detail. They outline:

- Definitions peculiar to their view of programming (see Appendix A)
- "The Program of Requirements" (Design Brief)--where they identify the two main functions of the design brief, identify what it should contain, and state the objectives
- Suggest a format for the design brief
- Identify the importance of updating the design brief if (and when) changes occur that affect the content of the program
- Identify the function, content and application of the "Requirement Stage" of the job (according to RAIC: "the stage where the client identifies a potential project, collects pertinent data, prepares his program of requirements and selects the Architect".

The RAIC methodically identifies the various phases of program development, and offers a suggested format for presenting the program.

2.2.7.3. Royal Institute of British Architects

The Royal Institute of British Architects (RIBA) offers comprehensive guidelines for their architects and planners (1965, note that the current literature was not available for review, but the review of the 1965-67 editions of "The Project (Planning)-2, The Techniques and Methods" vol. 4, part 3 handbook showed a greater level of concern about programming than the AIA displayed). A significant aspect of this work was the British method of presenting the programme (product) in two levels of detail (first brief and second brief). The significance of this distinction is discussed as the two levels of detail in Chapter 3 of this study.

2.3. PRODUCT MODELS

Product models represent complex information about something physical, i.e., a building, by simplifying the elements that comprise the whole. These models were initially developed for the AEC industry in the early 1980s. Khayyal (1990) researched various product models (i.e.; General AEC Reference Model (GARM), RATAS, Turner's building system model, Martin's distribution systems model, etc.) and identified the aspects of current AEC product models that were relevant to his master builder's information framework for project developers. The product model in this thesis is based on that research, and is related to research entitled "An Information Framework for Facility Operators" by Beckett (1991).

2.3.1. Khayyal's Product Model Architecture (PMA)

The Product model Architecture developed by Khayyal (1990) depicts two different types of information: building levels and discipline breakdown. The model sought to identify the generic information necessary to describe building components (products) in increasingly greater levels of detail. Khayyal identified five attributes that further describe the components of a facility. The attributes (form, function, economy, time, and mechanism) also identify the relationship between the building levels and disciplines (see Figure 2.2). Khayyal studied a master builder's viewpoint. However, another researcher in the Computer Integrated Construction program at Penn State, Beckett, considered the facility operator's perspective.

2.3.2. Beckett's Facility Operator Information Framework

Beckett identified the information needs of a facility operator (using a major university's facility management program in his case study). He considered the various AEC product models, and used aspects of the PMA in conjunction with the Construction Specification Industry's classification system to develop a Facility Operator Information Framework (FOIF). The FOIF is shown as Figure 2.3. The FOIF used an address coding scheme (comprised of "system," "level," "vantage," and "index") to access information needed by a facility operator. An "information code" is also incorporated that identifies the type of information an operator might need to access. For

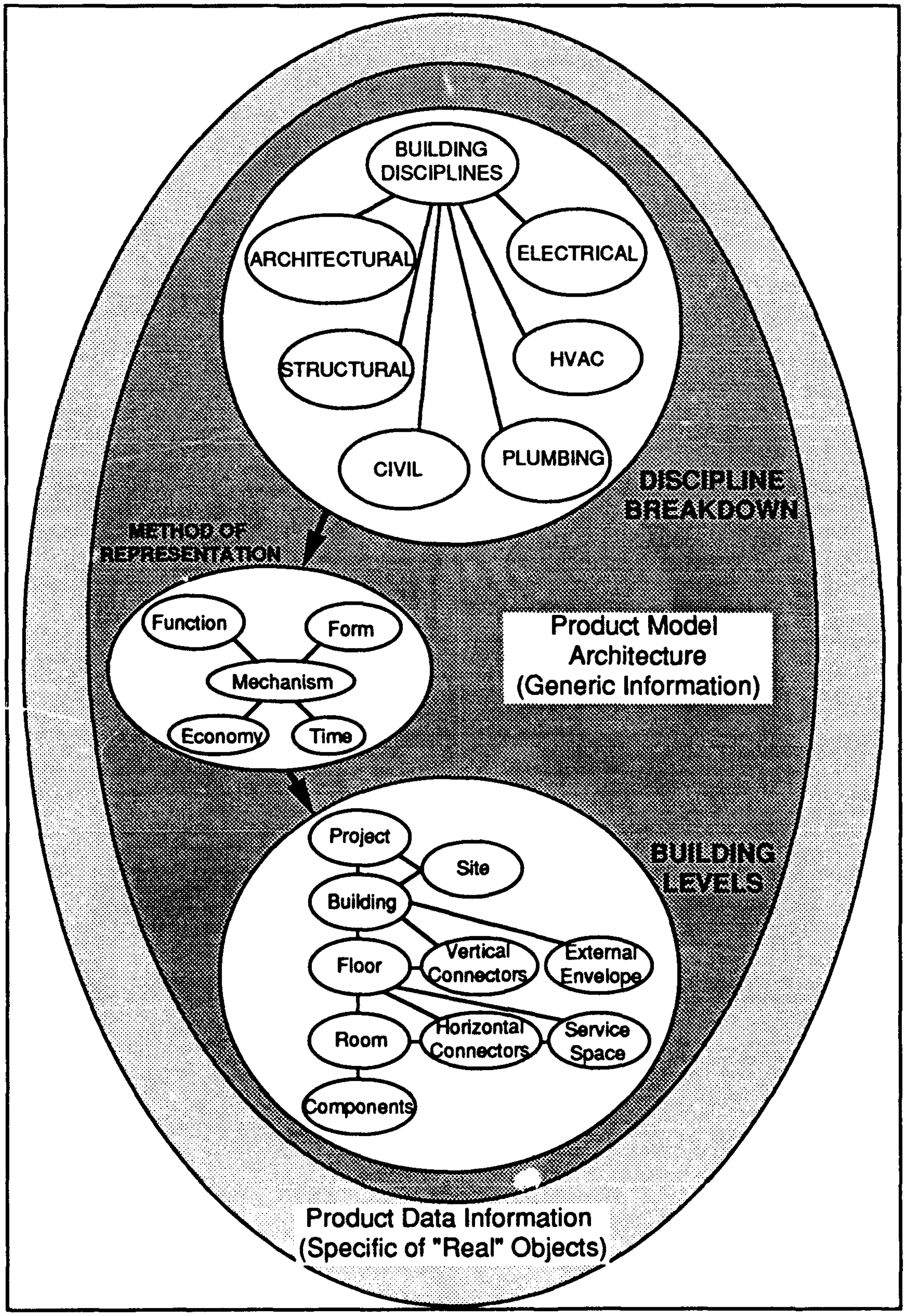


Figure 2.2: Khayyal's Product Model Architecture [1990, p. 79]

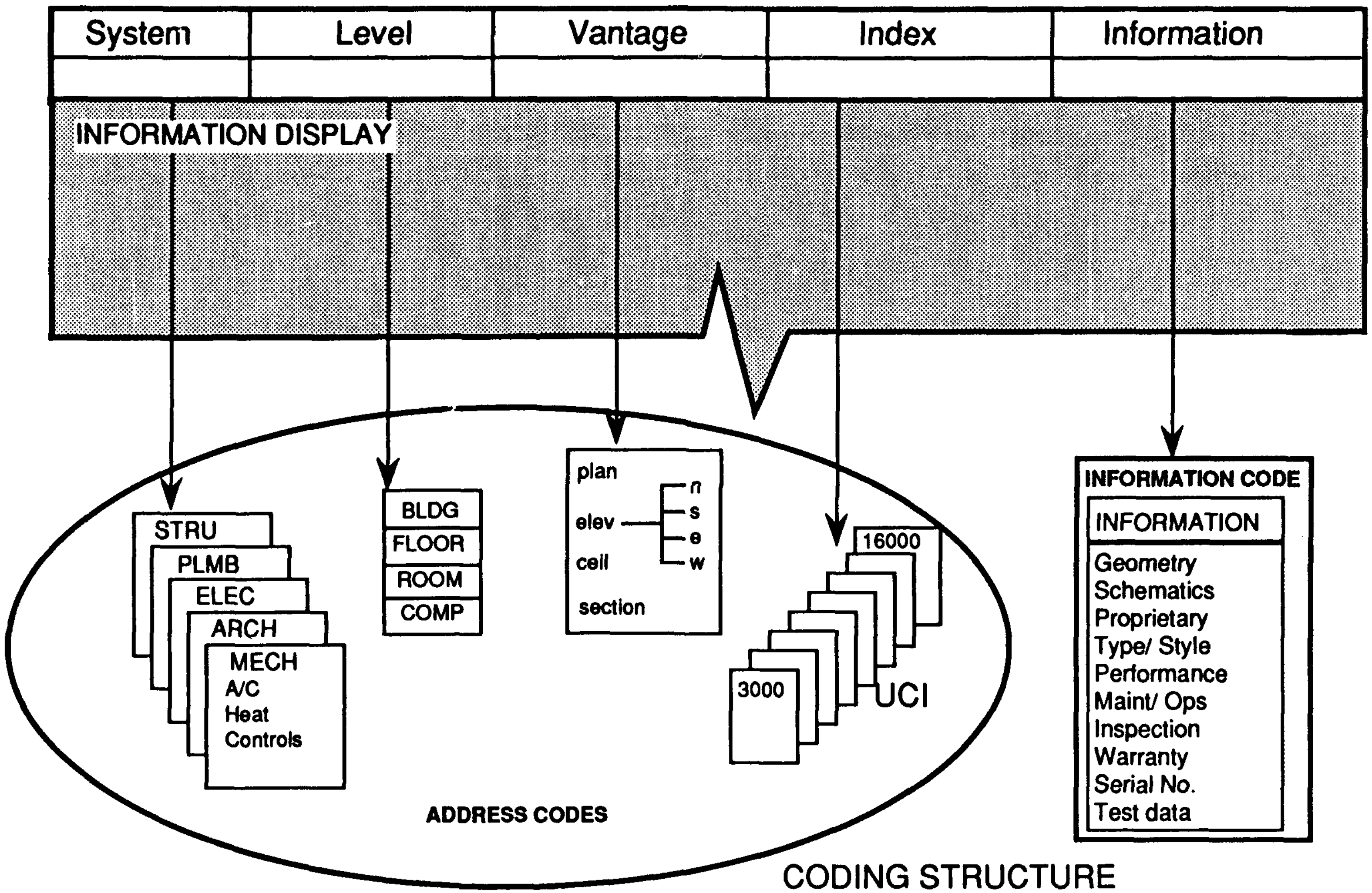


Figure 2.3: The Facility Operator's Information Framework (FOIF)

example, the warranty information or test data for building equipment items fall into this category.

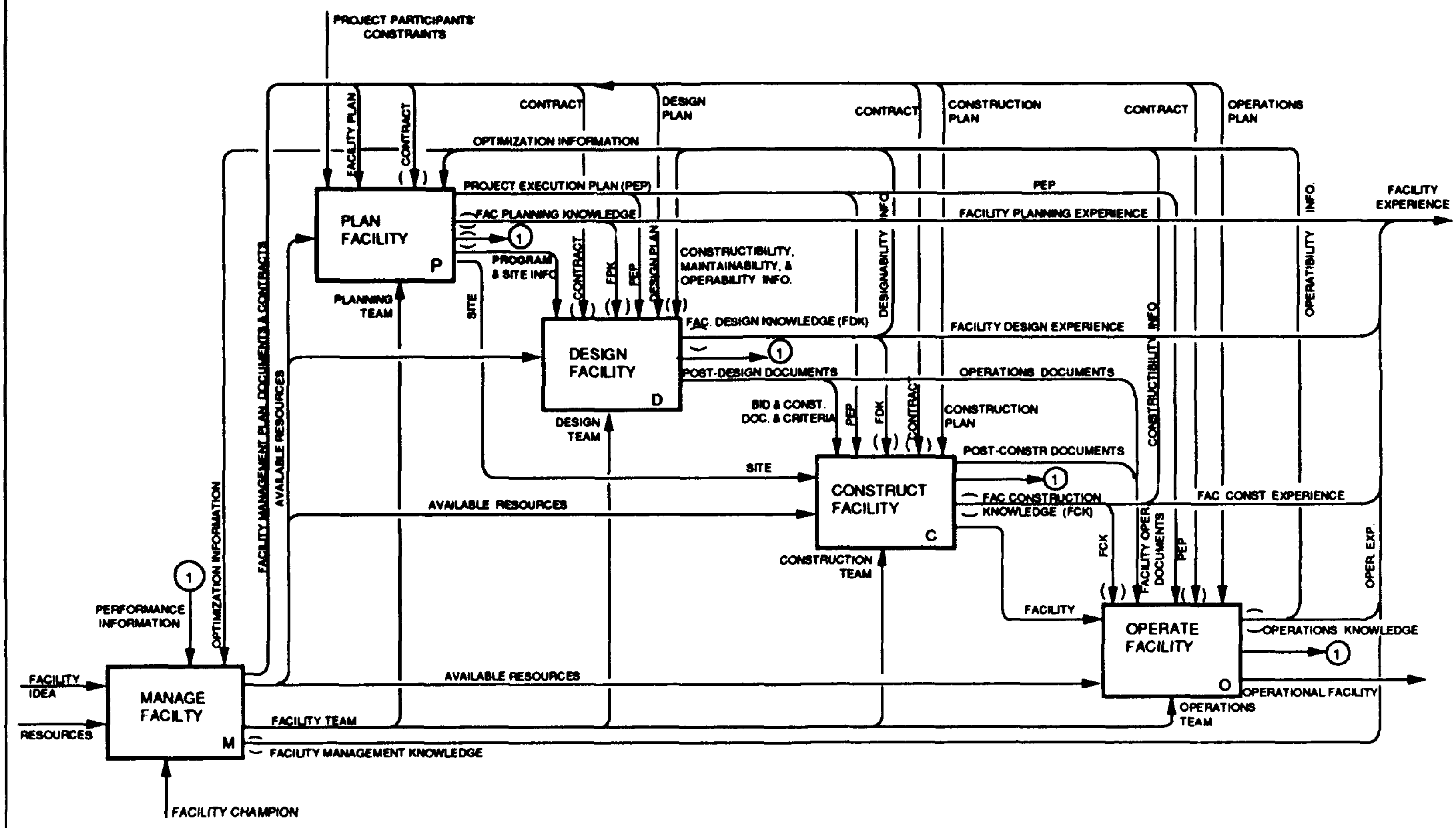
2.4. LIFE CYCLE OF PROVIDING A FACILITY

According to Sanvido (1990a) the life-cycle of the construction process encompasses the following processes: manage, plan (program), design, construct, and operate. This context of the facility life cycle was outlined in the Integrated Building Process Model (IBPM). The term "construction life cycle" refers to the entire process of providing a facility; it doesn't refer to the construction phase only.

The IBPM is significant to this thesis because it clearly describes the full life-cycle of construction in both process and product terms. Figure 2.4 shows the process depicted by the IBPM. Figure 2.5 represents a simplified view of the IBPM, where the product is highlighted. In this context, the program is the product of the planning phase of construction. The program is used to develop schematic and detailed design documents. The program could also be used as a standard to compare the performance requirements identified initially (in the program) against those found in the design, construction, and operations phases of the work.

2.5. RELATIONSHIP BETWEEN PROCESS AND PRODUCT

The schematic representation of the programming process and product (Figure 2.6) shows the product as an input to the next phase of the



NODE: F.	TITLE: PROVIDE FACILITY	BY:	CHECKED BY:

Figure 2.4: Level 2--Components of Provide Facility (Sanvido, 1990c, pg. 159)

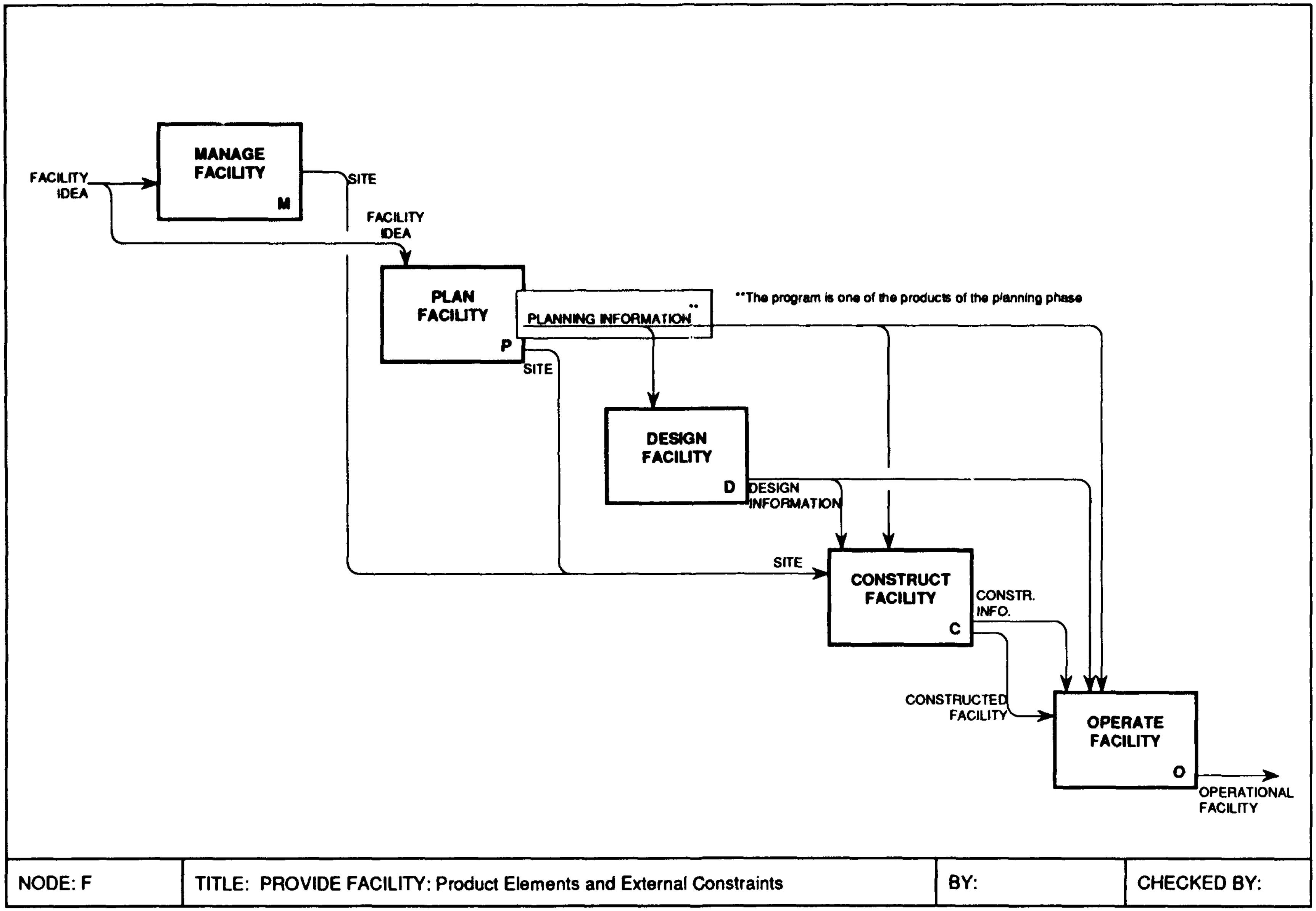


Figure 2.5: The "Product Elements" of the IBPM (Sanvido, 1990c, pg. 162)

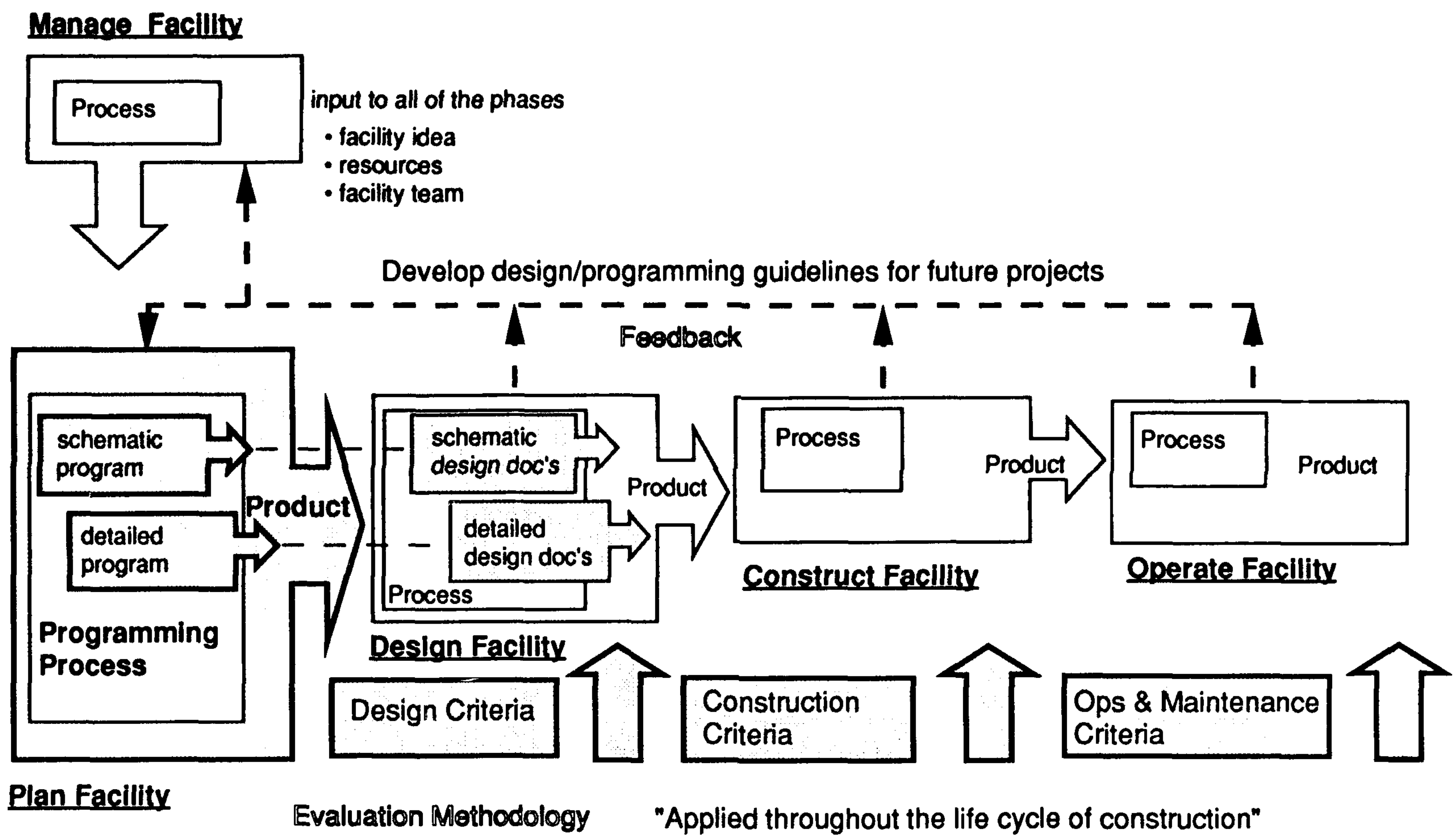


Figure 2.6: Schematic Representation of the Programming Process and Product

process. This is consistent with rules established for the IBPM (Sanvido, 1990a). The schematic figure also identifies the two levels of programming and design detail suggested by Peña (1987). The purpose of Figure 2.6 is to show the relationship between process and product.

The product of a given phase becomes the input to the subsequent phase. For example, the program is a product of the planning phase, and provides input to the design phase. The product is a physical and tangible link between the phases. The process that leads to the product is less tangible; but the quality of the process is often reflected in the product.

2.6. DISCUSSION--CURRENT INDUSTRY NEEDS

The AIA does not provide clear guidance related to either process or product aspects of programming (even though the AIA Press has published work on programming). Also, the AIA standard contract documents stipulate that the owner shall provide the program. However, this requirement is unrealistic, because most owners do not have the training or experience to compile a program. As a result, the architect may inevitably program the work, without adequate compensation (resulting in placing a low priority on both the programming process and product).

There is no clearly established method of gathering, storing, retrieving and updating the programming information for a facility proiect. While some firms have developed standards of practice for developing and presenting program information, this is the exception, not the rule.

There is no procedure or methodology for using the performance criteria established in the program to evaluate subsequent phases of the life cycle. Post occupancy evaluations (POE) are being conducted by various firms (Victoria University, Daish et al. ,1982; Preiser, 1982; the National Building Technology Centre, Boyd and James, 1988; among others) but POE focuses on feedback (to the program) only after construction is complete--not before.

2.7. SUMMARY

Programming is a unique and distinct aspect of the planning process. As such, it has its own terminology (which is presented in Appendix A for clarity). The process of developing a program, then presenting it to the owner (and other members of the facility team) is important, however this thesis focuses on the program as a product.. The program is the foundation for design development. It is also dynamic, and may require revisions throughout the life cycle of construction. Currently, the AIA doesn't present guidelines for handling the process and product of programming. Therefore, there is a need to develop an information framework suitable for both. Chapter 3 describes that information framework.

Chapter 3

THE FACILITY PROGRAMMING PRODUCT MODEL

3.1. THE FACILITY PROGRAMMING INFORMATION FRAMEWORK

This chapter develops the criteria for structuring the information framework and describes the FPPM development. The various stages of refinement to the model are identified. Lastly, the structure and rules for using the final version of the FPPM are discussed.

3.1.1. Criteria for the Framework

The criteria that should be satisfied to meet the information framework requirements of a facility programmer are described below. The criteria are based on a combination of the criteria for Khayyal's Product Model Architecture (1990) and the programming literature (as discussed in Chapter 2). For each criterion, the criterion is stated, then it is discussed.

1. **Consistent Framework**: Provide a consistent structure to contain programming information. The framework must provide the ability to gather, store, retrieve, and update information. The

framework in the FPPM was based on similar structures established by Khayyal (1990) and Beckett (1991).

2. **Open Framework**: Information related to specific categories (at various levels), would be applied to a given project as the project's information needs dictated. For example, if "social issues" information related to user behavior patterns isn't available, or needed, by the facility team, that portion of the framework would not be utilized. Only information needed by the facility team would be applied to the framework.

3. **Comprehensive**: The product model must be able to handle any of the information elements required by the facility programmer. The types of information (elements of the model) must be established to account for any product information requirement.

4. **Evaluation and decision making tool**: The program must be useable as a vehicle to analyze the priority and value of the building requirements.

5. **Accessible** later in the life cycle: The FPPM must provide a product that has the capacity to carry information forward to each stage of the life cycle. The FPPM should contain information the owner's representative needs to effectively communicate with other members of the facility team, resulting in satisfying the owner's facility goals and requirements.

6. Contain only **essential information**: Avoid "data clog". When gathering information, discard nonessential information. Instead, use only information which is essential to a given building level or system

in a given general or specific programming information category. Peña (1987) calls this "getting to the essence."

3.2. DEVELOPMENT OF THE FRAMEWORK

This section describes the evolution of the FPPM through three successive generations. The initial conceptual model (FPPM1) was based on the literature and the author's experience in facility programming and incorporated the performance criteria above. The second generation FPPM (FPPM2) was developed after reviewing 15 sample programs to see if various products (programs) had common characteristics, format, or information elements. The third generation model (FPPM3) resulted from refinements made after a review by industry experts. The initial stages of the model's development and refinement are not included in detail in this chapter for the purpose of brevity. The process is briefly outlined below.

3.2.1. Conceptual Basis for the Model

Khayyal's product model (1990) was refined by Beckett (1991) to include information needed by a facility operator. Khayyal's and Beckett's models were compared to programming information requirements. Then, programming specific information categories were included in the FPPM structure. Information not needed in the program was not considered.

3.2.2. The First Generation FPPM (FPPM1)

The FPPM1 included information at a level of detail which was too specific to be useful in the initial planning stages. For example, it contained the building level information shown in Khayyal's PMA (1990) (e.g. project, building, floor, room and component were all levels of detail proposed by Khayyal). This building level code allowed more detail than is required by a program. This was changed to a binary code, called "level" in the FPPM, which maintains what Peña and RIBA discuss as two levels of programming information (initial/schematic and detailed).

There were also eight categories of programming information in the FPPM1. "Historical" information was changed to "preprogram" to improve clarity. The categories "Behavior" and "Values" were consolidated into a "Human Factors" category in the FPPM2 (this category also changed later).

3.2.3. The Second Generation FPPM (FPPM2)

The FPPM2 was modified after reviewing 15 facility programs. The study was conducted by this author in May, 1991, and was entitled "A Summary of Program Evaluations". This section introduces the nature of the analyses of those facility programs and educational specifications at the summary level and detailed level. First, the scope and objectives of that study are outlined. Concluding remarks then summarize the important realizations in the study.

3.2.3.1. Scope

The programs were analyzed in order to provide a basis for the review and refinement of the FPPM1. The program analysis occurred in two stages. After the 15 programs were reviewed, nine were selected to receive a summary level review. Then, three recommended formats (ways to organize the program) were shown on the analysis matrix (so they can be compared to the programs). Afterwards, two of the best programs were reviewed in detail. The objectives of the analysis relate to the development and refinement of the model.

3.2.3.2. Objective

This study had three primary objectives. These objectives were satisfied in the study and are stated here:

- Test the FPPM1--provide a basis for the development and refinement of the FPPM1's structure (framework) and rules for use.
- Provide familiarization with industry standards for programming (format, content, etc.).
- Test commonality--determine what (if any) common characteristics various programming documents possess.

3.2.3.3. Summary of Observations

The FPPM2 was simplified by combining information relevant to both the *human behavior* and *values* categories into one category called **Human Factors**. Often, information related to either human behavior or values really fits into both categories, so consolidation makes sense.

Programming information is developed in the planning phase, and relates most strongly to the design phase. However, using the program to verify performance criteria, or to ensure the program is updated as changes are made throughout the life cycle is still valid.

Programs reviewed consistently contained level-one (schematic) programming information. Some programs were developed and presented in more detail (level-two). The increased level of detail is appropriate to some situations (otherwise the programmer would not have spent the time and energy to gather, analyze and present it). This aspect of the model should remain, and the product should reflect the level of information required by the members of the "facility team".

3.2.4. The Third Generation FPPM (FPPM3)

The FPPM2 was presented to three industry experts for their review and comment. The professionals reviewed the model, and the rules for using it. Refinements were then made to the model based on feedback from the programmers and designers

Each professional was (or is currently) affiliated with teaching programming and/or design; and has at least 20 years of practical work experience in the profession. A sample of the questions asked, and feedback received during the interview portion of the industry review is presented in Appendix C. The next section identifies the elements of the FPPM and describes the rules for using the model.

3.3. ELEMENTS OF THE PRODUCT MODEL

The coding scheme and information display of the FPPM is presented as Figure 3.1. The model contains a series of categories of information, codes, and "cells". The "cell" represents the area in the framework where either a category, code or "information display" would be represented. The coding scheme allows the user to access information in the model, or to organize the presentation of the information contained in the model. A sample of how the coding scheme would work is shown and the coding schemes are described in the next section.

3.3.1. Coding Scheme

The FPPM defines a coding system, as well as an information display for 6 types of programming information. The classification system has two parts: 1. the address code--which acts as an "address" to categorize information; 2. the utility code--which represents the priority of an element of information relative to the project. Each coding scheme is described below.

3.3.1.1. Address Coding Scheme

The address codes are level, general categories of programming information, system and graphic link (see Figure 3.2). The address coding scheme represents a way to access (input or update) information in the program by identifying the level, type, or discipline related to that information. A description of the four address codes follows.

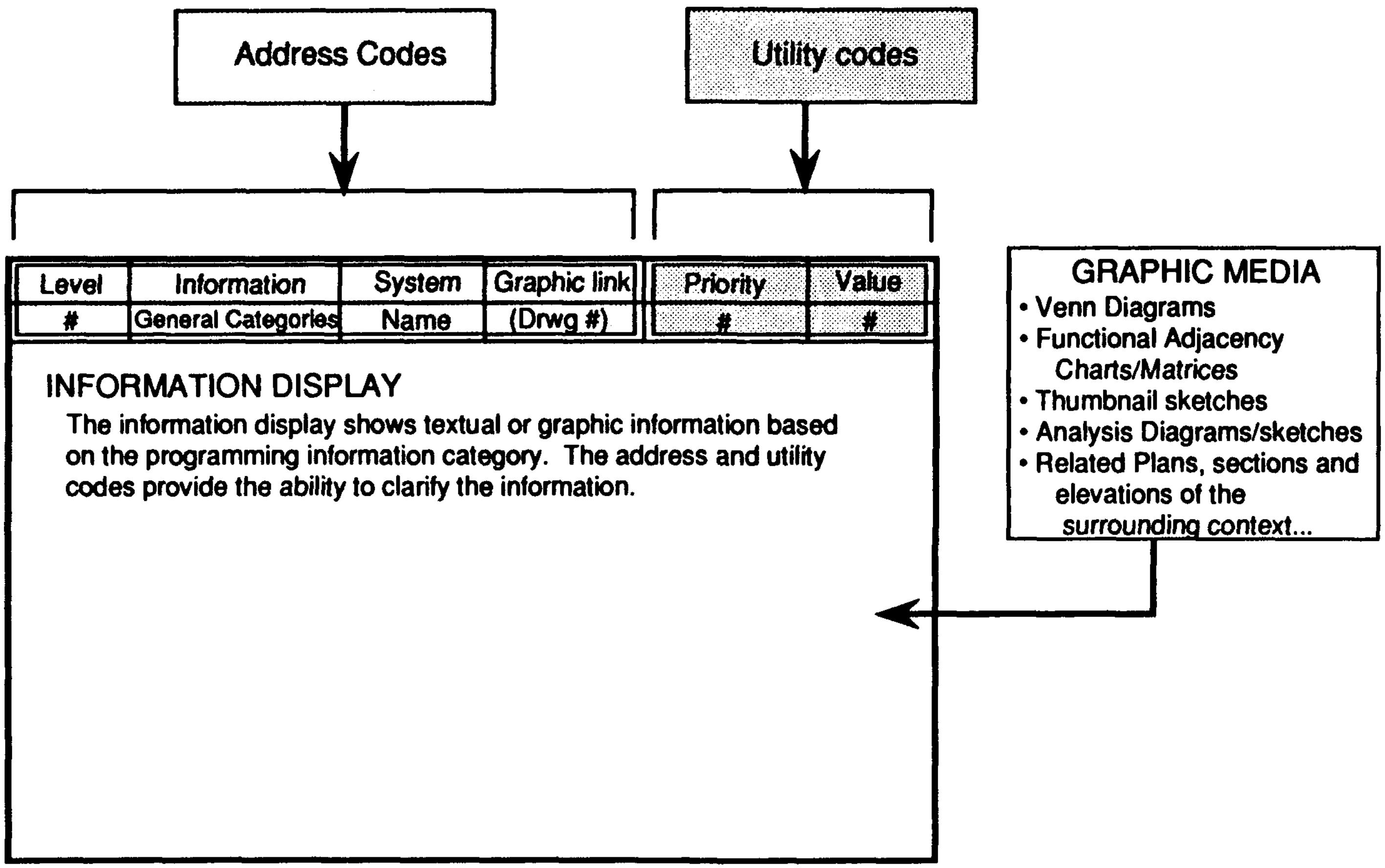


Figure 3.1: Elements of the Facility Programming Product Model

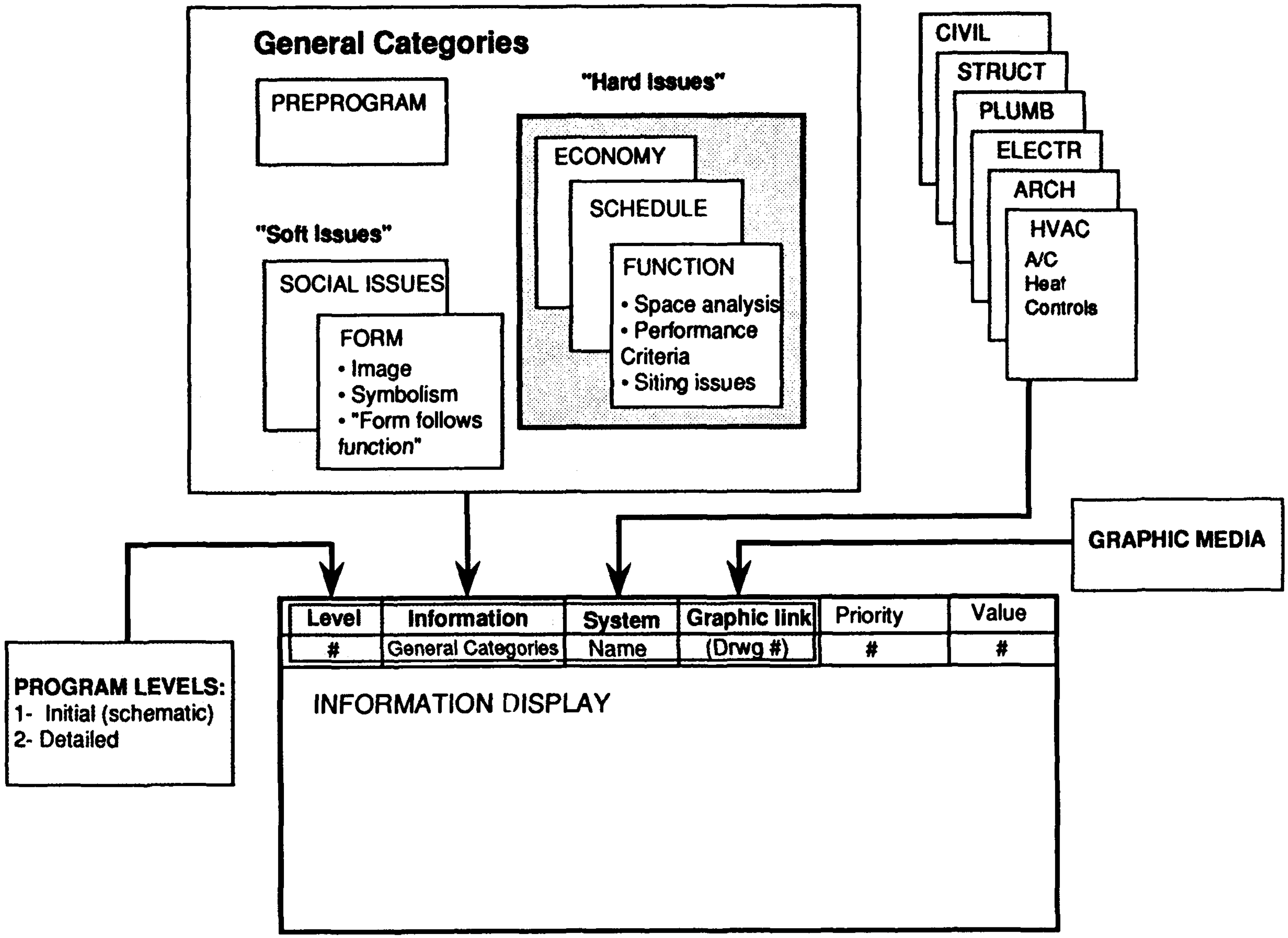


Figure 3.2: Elements of the FPPM Address Coding Scheme

- <u>Level</u>--defines the level of detail demonstrated by the program information. It is shown as level 1-initial/schematic or level 2-detailed. For instance, the quantity and type of special equipment in a room would be level 2 information.
- <u>General Categories</u> of programming information used in the FPPM are broken into three categories: "hard issues", "soft issues" and "preprogram issues". Hard issues are objective in nature. The hard information categories are:

1. Economy: The efficient and sparing use of the means available for the end proposed. Economy implies an interest in achieving maximum results from the initial budget and the maximum cost/effectiveness of the operation and life cycle costs. (Peña, 1987)
2. Schedule: The project schedule, or time lines. This also deals with the influence of history, the inevitability of change from the present and with projections into the future. (Peña, 1987)
3. Function: How the design product will work to assist in the performance of the job it is intended to support. Function is also the way people and things will move about to complete the tasks they have to do (Peña, 1987). Some examples of functional issues follow:

Performance Criteria is an element of function defined as--Those requirements stemming from the unique user needs in terms of the physical, social and psychological environment to be provided.

These will involve the adequacy, the quality and the organization of space (Peña, 1987).

Code issues: those regulatory requirements which must be satisfied to protect human safety and in order to obtain project approval.

Soft issues are subjective in nature. The soft information categories are social issues and form. Each is described below.

4. **Social Issues**: The various demands that society places on a project comprise the social issues. Some examples of this type of information follow:

Behavioral Factors: Those requirements stemming from the generalized human needs in terms of the physical, social and psychological environment to be provided. These human needs involve such general categories as self-preservation, physical comfort, self-image and social affiliation--initially expressed as specific goals.(see human requirements, Peña, 1987).

Political/Community issues: Those requirements identified by the local "community" (or board of directors) that must be considered to obtain necessary approval(s) for funding, siting, zoning, etc.

Architectural values: As they relate to architecture, are categorized as being one of three types (Hershberger, in Programming the Built Environment (Preiser), 1985): "enduring values" (firmness, commodity, and delight), "institutional values" (continuity or history of development), and "circumstantial values" (i.e., environmental--site, climate, urban context, etc., societal--cultural, legal, community, etc.). Do not confuse this definition of values with the "value" cell of the FPPM.

5. Form: In design, form means the shape and structure of a building as distinguished from its materials. In programming, form refers to what you will see and feel, avoiding the suggestion of a design solution. It's the "what is there now" and "what will be there." (Peña, 1987)

The last category is "preprogram information", which can be a combination of hard and soft issues; it is critical to the success and validity of the program. A description of preprogram information follows:

6. Preprogram Information: Information relevant to the programming of a facility that was assembled *before* the program was initiated, but is instrumental in supporting or clarifying elements of the program. Examples are: feasibility studies, site selection studies, master plans, etc.

Specific programming categories representative of those associated with each general category above are shown on Figure 3.3 and 3.4. The listing of specific categories are presented based on the site and/or building level information.

"SOFT ISSUES"

SOCIAL ISSUES

BEHAVIORAL FACTORS

building

- SERVICE GROUPINGS
- PEOPLE GROUPINGS
- ACTIVITY GROUPINGS
- PRIORITY
- SECURITY CONTROLS
- FLOW (SEPARATED VS. SEQUENTIAL)
- RELATIONSHIPS
- ADAPTABILITY
- CONVERTABILITY
- SAFETY/HEALTH
- DURABILITY/EFFICIENCY
- UTILITY/AESTHETICS

site

- SEATING ARRANGEMENTS
- CIRCULATION
- TEXTURES/PATTERNS
- IMAGE
- ORIENTATION
- SAFETY/HEALTH
- DURABILITY
- EFFICIENCY
- UTILITY/AESTHETICS

COMMUNITY ISSUES

site

- ZONING
- PEDESTRIAN AND VEHICULAR TRAFFIC
  - existing
  - proposed
- SURROUNDING CONTEXT
- SOCIAL "CLIMATE"
- FUTURE GROWTH AND POTENTIAL FOR CHANGE

VALUES

- ENDURING
  - "survival" (firmness)
  - "good life" (commodity)
  - "meaning and art" (delight)
- INSTITUTIONAL
  - continuity or history of level
- CIRCUMSTANTIAL
  - environmental
  - human
  - societal
  - temporal (permanence)
  - economic
  - aesthetic

PREPROGRAMMING INFORMATION

STRATEGIC LEVEL PLANS

site

- SELECTION CRITERIA
- MOVEMENT:
  - vehicular
  - pedestrian
- ZONING CONSTRAINTS
- CODE ISSUES
- PARKING
- ENVIRONMENTAL FACTORS

building

- POPULATION PROJECTIONS
- MASTER PLANNING ISSUES
- DEVELOPMENT ISSUES
- REVIEW OF PREVIOUS RECOMMENDATIONS

past

- HISTORIC PRESERVATION
- STATIC DYNAMIC ACTIVITIES

FORM

site

- relationship to site
- entry (orientation/emphasis)

facility

based on:

- QUALITY
  - projected image
  - spatial/Technical/functional quality level
  - cost (bldg., equipment)
- "FORM FOLLOWS FUNCTION"
- IMAGE
- SYMBOLISM

Figure 3.3: Specific Programming Information Categories--Soft Issues and Preprogram Information

FUNCTION

site

- PEOPLE
  - movement on the site
- ACTIVITIES
  - public activities
  - green spaces/plazas
- RELATIONSHIPS
  - to the building
  - to other sites/buildings

space analysis

- PEOPLE
  - mission
  - maximum number
  - individual identity
  - space requirements
  - grouping
- ACTIVITIES
  - security
  - progression
  - type/intensity
- RELATIONSHIPS
  - segregation
  - information exchange
- MATERIAL MOVEMENT
- WORK NODES

performance criteria

- AREA
- CIRCULATION
- AMBIENT ENVIRONMENT
- SAFETY AND SECURITY
- SURFACE TREATMENTS (safety, cost, image, direction finding)
- FURNISHINGS
- FLEXIBILITY
- SYSTEM PERFORMANCE (specifications/criteria)
- OPERATING REQUIREMENTS
- MAINTAINABILITY

ECONOMY

- INITIAL COSTS
  - parameters
  - budget
  - market analysis
- OPERATING COSTS
  - cost effectiveness
  - energy costs
- LIFE CYCLE COSTS
  - maximum return
  - return on investment
  - reduce life cycle costs

"HARD ISSUES"

SCHEDULE

- PRESENT
  - activities (vs. future)
  - adaptability
- FUTURE
  - time/cost
  - change
  - growth
  - occupancy date
  - phasing
  - convertibility
  - expansibility

Figure 3.4: Specific Programming Information Categories--Hard Issues

- <u>System</u>--defines the discipline involved in a particular aspect of the program; (i.e., civil, architectural, electrical, etc.) Each of the disciplines involved in the building process would have a need for information related to their discipline. This code allows for information to be sorted and presented according to which design, construction or operations professional needed it for a given discipline.
- <u>Graphic link</u>--a number that relates a concept to a graphic image (drawing #).

3.3.1.2. Utility Coding Scheme

The utility coding is a way to show the importance that information has to the programmer (and other members of the facility team). There are two codes shown in the model (see Figure 3.5). The first represents the priority of the information. The second identifies its value. Each is described below:

- <u>Priority</u>--the priority is categorized as being in one of four levels and represents how critical the information is to the success of the project as a whole. The categories are described below:

  1. Mission Essential: This element of the program must be satisfactorily identified, understood, and included in the program for the facility to be usable by the organization. In other words, the organization can not function as they need to if this aspect of the program is not satisfied.

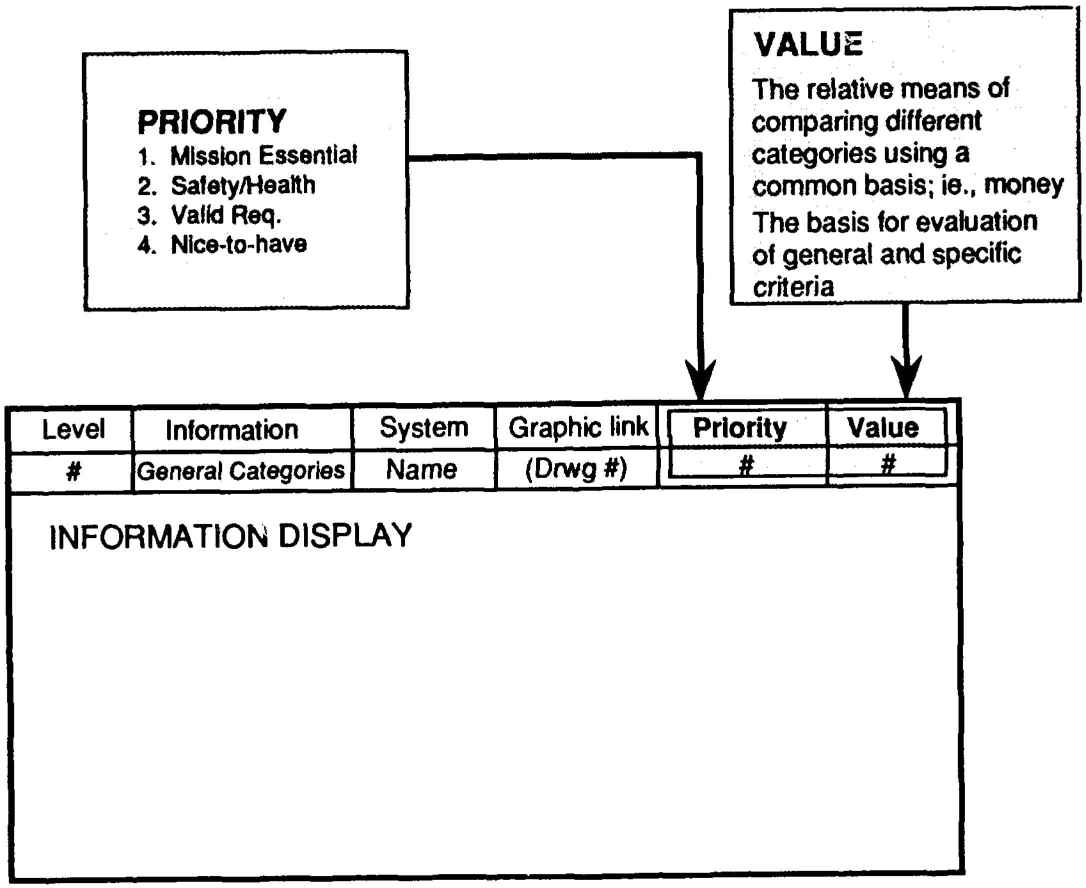


Figure 3.5: Elements of the FPPM Utility Coding Scheme

2. Safety/Health: This element of the program should be resolved in order for workers, visitors, or people passing by the facility to be safe. Life-safety code issues are excellent examples of information having this priority.

3. Valid Requirement: These requirements are bonafide "needs" (as defined in Appendix A), but they don't fall into one of the first two categories. The impact of not providing these is not as significant as it would be with either mission essential or safety/health requirements.

4. Nice-to-have: These elements of the program are just as the title implies. These requirements fall into the "wants" category, as it is defined in Appendix A.

- **Value**--a relative means of comparing different categories using a common basis. For example, when soft issues (like social issues) are compared to hard issues (like schedule) there needs to be a common basis for comparison. The recommended basis (scale) for comparison is cost. The cost should be based on a given date. Cost can either represent the cost to provide something (i.e., the cost of buying air conditioning equipment), or the cost of not providing something. The value cell would be used as a "tie-breaker" when analyzing two types of information that have the same **priority** level. For instance, take the political decision regarding how much of the open athletic field should be consumed by the building foot print. Politics demand that the building occupy very little of the site. Compare the cost of a larger building foot print , the cost of adding another story to the facility, and the future cost (impact) of not providing the space. Now, the decision making body can compare the information on common terms and can decide which alternative to select, based on the cost and political ramifications associated with each alternative. This process demonstrates how the value cell can be used to support the evaluation of different "trade-offs".

3.3.1.3. Information Display

This field in the frame would display any type of information related to the information shown in the coding structure. In essence, this "cell" is part of the skeleton to the framework, and the meat is what is shown in the information display. The information could be graphic (a sketch, schematic, bubble diagram, etc.) or textual. The composite view of Figures 3.1-5 is shown as Figure 3.6.

The following examples outline the type of information that might be found in the various cells, and are reinforced by a sample sketch of what information might appear in the "information display" area. Figure 3.7 shows the coding scheme for schematic information (level-one) and a sample display in the information display field. In this example, only the address codes are utilized for this particular information related to the "form giving" considerations on the project. A "1" would be assigned to level and "Form" would be assigned to the general category of programming information. The system is "Architectural" and the graphic link would retrieve a sketch of the relationships between the elements that contribute to the form of the facility. Please note that the arrows shown on Figure 3.7 represent a link to other information, but assume no directional role.

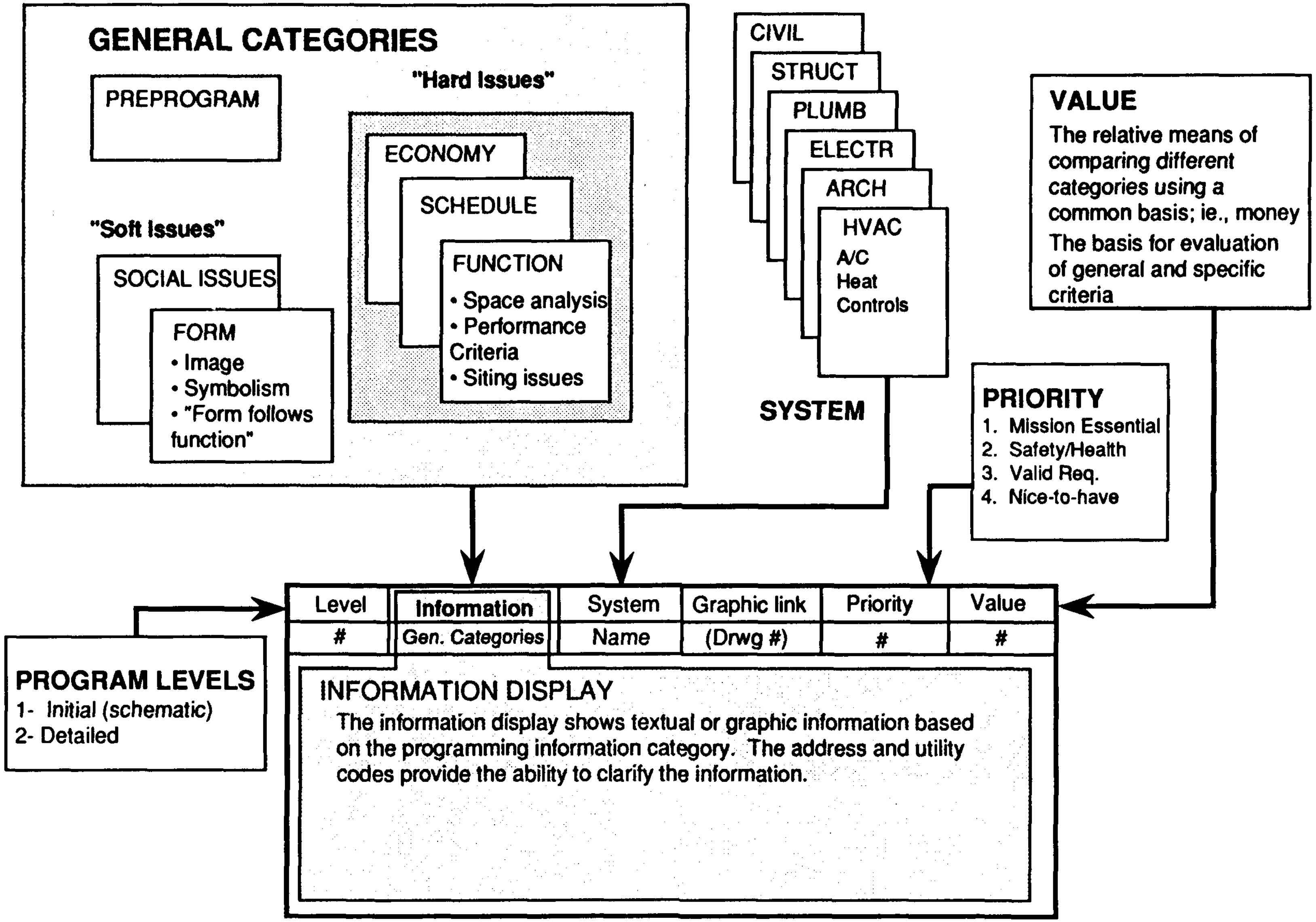


Figure 3.6: Composite View of the FPPM

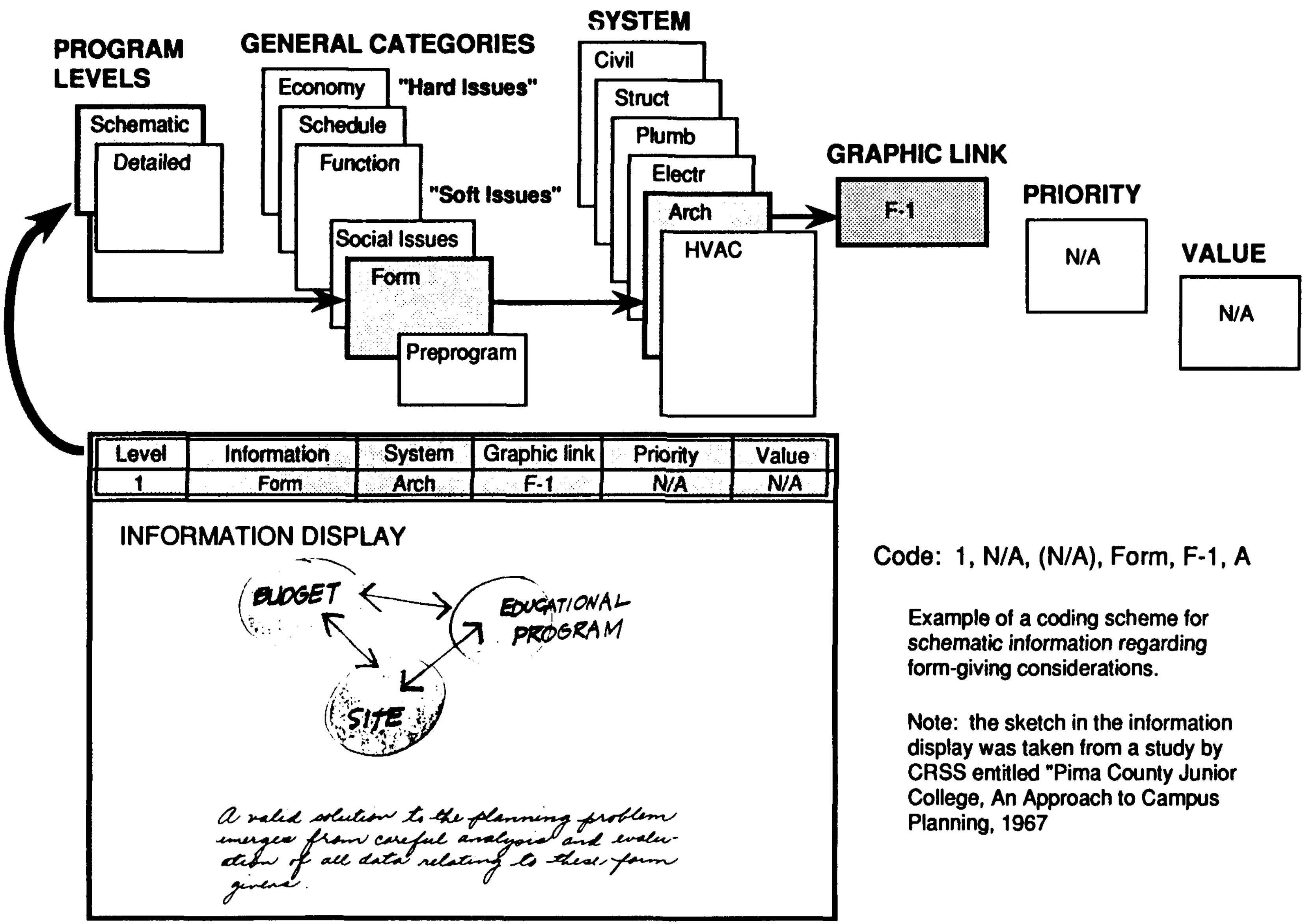


Figure 3.7: Sample, Level 1 FPPM Coding Scheme

Code: 1, N/A, (N/A), Form, F-1, A

Example of a coding scheme for schematic information regarding form-giving considerations.

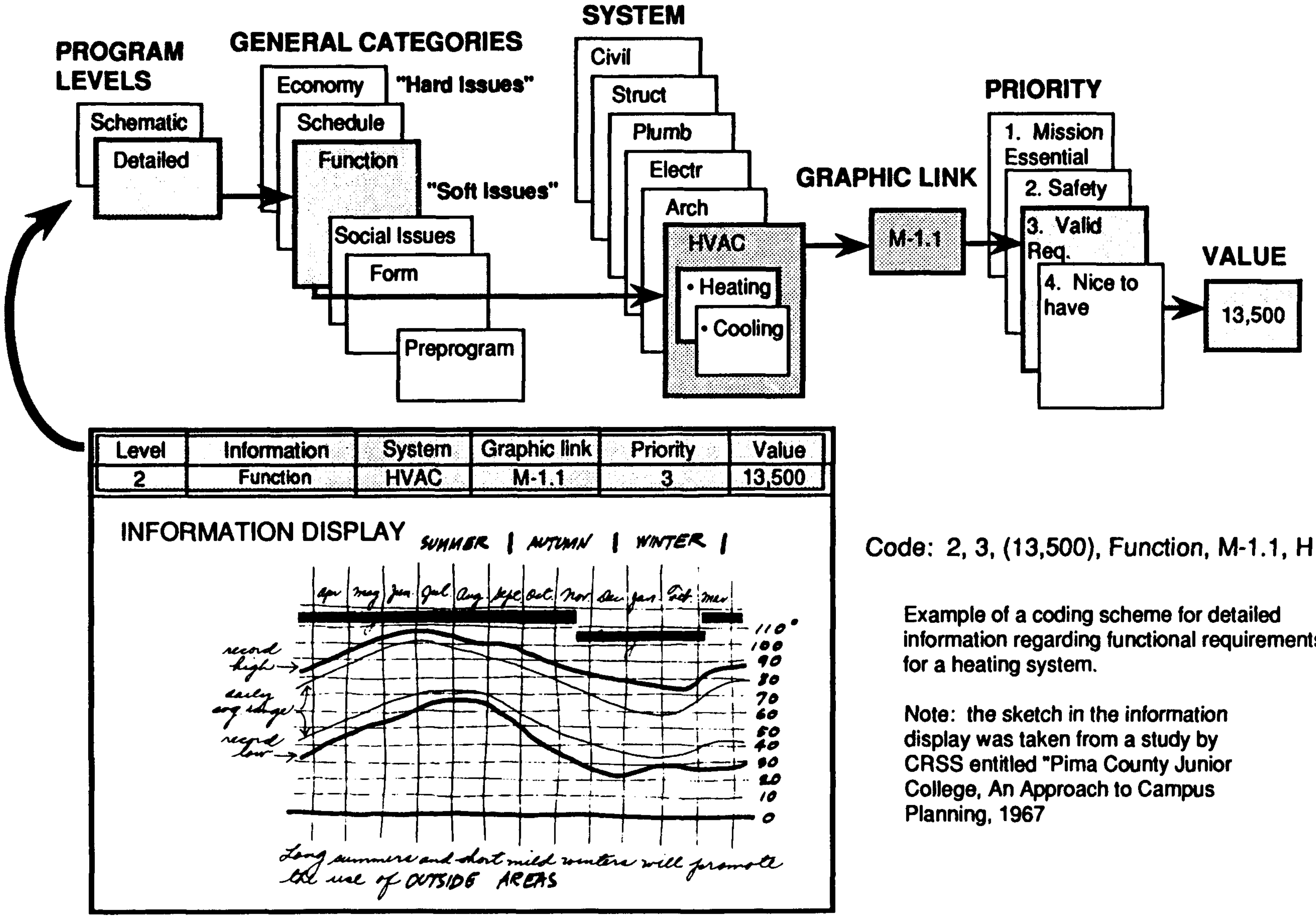
Note: the sketch in the information display was taken from a study by CRSS entitled "Pima County Junior College, An Approach to Campus Planning, 1967

In another example, if the user wanted to retrieve detailed information about the functional requirements for a heating system, he/she would assign a "2" for the level, "function" for the general category of information, and "HVAC" for the system. The system has a hierarchy to show the various types of HVAC systems available. Heating would be selected from those options. The function aspect of the general programming information would also be selected (see Figure 3.8). There are various specific-information categories associated with the function of the system. The result of this inquiry through the address codes might result in a schematic representation of the performance criteria for the heating system. Any graphic media related to the information could be accessed through the graphic link.

3.4. CONSIDERATIONS WHEN USING THE MODEL

Effective communication with members of the facility team is a critical aspect of gathering information, storing it, and making subsequent retrievals or updates. In this section, the general rules for using the model are presented. Afterwards, the guidelines for gathering, storing and retrieving information are identified. The general rules for using the model are:

- Programming is a two phase process. Initially, schematic information is gathered, organized and presented. Then, additional details are included as the program develops.
- Distinguish between "wants" and "needs" (Peña, 1987). The priority code in the FPPM allows the user to assign a relative priority to the



Code: 2, 3, (13,500), Function, M-1.1, H

Example of a coding scheme for detailed information regarding functional requirements for a heating system.

Note: the sketch in the information display was taken from a study by CRSS entitled "Pima County Junior College, An Approach to Campus Planning, 1967

Figure 3.8: Sample, Level 2 FPPM Coding Scheme

information, based on how important the information is relative to the facility team's needs.

- Use the FPPM as a checklist to see if all of the general (and specific) programming information requirements were considered at the appropriate level of detail for the facility team.
- Avoid "data clog" by discarding nonessential information.

These guidelines for using the model should be considered when gathering, storing, retrieving, analyzing, or updating information.

3.4.1. Gathering Information

Initially, gathering information (in order to understand the nature of the facility requirements) is the primary task for the programmer. The model can be used as a checklist to see if all of the general (and specific) programming information requirements were considered at the appropriate level of detail for the facility team. The level of information is important, because the need to gather schematic versus detailed information will vary based on the contractual relationship between the designer and programmer and the nature of the building type being programmed. For example, a hospital will require a more detailed definition of the facility requirements than a single family residence.

The product model can serve as a reminder or guide in identifying what type of information should be gathered, or perhaps more importantly, identifying what program information elements have not been gathered. Ultimately, the facility team will decide what format the information should take, resulting in a program that meets their needs. The process of gathering information doesn't change with varying building types, even though the type and level of information probably will.

3.4.2. Storing Information

Information is stored in the model based on the code that ties it to a specific "cell" in the framework. The overall "address" for the individual frame is a summary of the individual codes that are related to specific programming information. The information codes are shown on Figure 3.1, and examples of the coding scheme are provided as Figures 3.7-8.

3.4.3. Retrieving Information

Accessing information in the framework is essential when developing, evaluating, presenting or updating the program. Once information is retrieved, it can be viewed, or updated (modified) as required. Here are some of the ways to retrieve information:

1. Enter the address of the information needed.
2. Go to the general programming information category related to the information, then search those fields.
3. Sort information according to discipline, then narrow the search by identifying the priority,level of detail, or general programming category.

3.4.4. Analyzing Information

The program can be a tool to assist the decision making process. As such a tool, the value of programming information carries great significance when using the model. "Value" is a way to compare soft issues equally with hard issues by developing a common baseline for the comparison. The value that a certain member of the facility team might place on a specific project issue can not be generally defined. However, the need to compare objective issues on a consistent basis with subjective issues is easy to recognize--consider the old adage "comparing apples to apples".

William P. Ross (see Appendix C) suggested that the common basis for comparison should be cost. His rationale is that cost is typically the lowest common denominator. For example, the cost of using brick versus a decorative concrete block for the exterior skin of a facility to improve the organization's image will allow the decision making body to compare the alternatives with other hard issues on an equal basis.

3.4.5. Updating Information

Updating the program with modifications to the facility requirements is a critical aspect of the overall process. The need to update the program is typically based on either changes to existing information, or new information.

To update information, you would first access the appropriate frame. Then, make whatever corrections, additions, or deletions are required. Only personnel granted access rights (by the owner or programmer) to the program would be able to make modifications. Modifications would record the date and time the change was made and the person responsible (and accountable) for the change.

3.5. SUMMARY

The model establishes the framework/structure for the program. It creates a systematic way to store, manage and retrieve programming information. The facility programming guide (presented in Chapter 5) uses a checklist format and is an extension (and further development) of the considerations for using the model. It enables the user (owner's representative) to:

1. Gather information for the program i.e., ensure relevant information related to the general program information categories is gathered, analyzed, evaluated, organized and presented to the designer.
2. Extract needed information from the program, i.e., test how well programming criteria is being met during subsequent phases of the life cycle (design, construct, operate). Here, the focus is on the relationship between programming and design; and regards how well the programming criteria is being met by the design solution.

The case study addressed how information was gathered, presented and evaluated during the life cycle of construction and is presented in the next chapter.

Chapter 4

FACILITY PROGRAMMING CASE STUDY

4.1. GENERAL DESCRIPTION

This chapter outlines the case study by describing the scope and objectives of the case study, then the methodology used in the research. The project familiarization and interview process and objectives are described. The role each owner's representative (OR) had in the execution of a project is then identified. Lastly, the findings of the case study research are presented.

The FPPM was applied during the case study portion of the research. The model was tested using case study data from four pairs of public sector projects in various phases of the construction life cycle. These projects were chosen in order to study how the programming information needs change over the life of a project. Each pair represented a phase of the project life cycle. Project size, complexity, funding, and method of programming were intentionally varied in order to represent the diversity of projects managed by this owner.

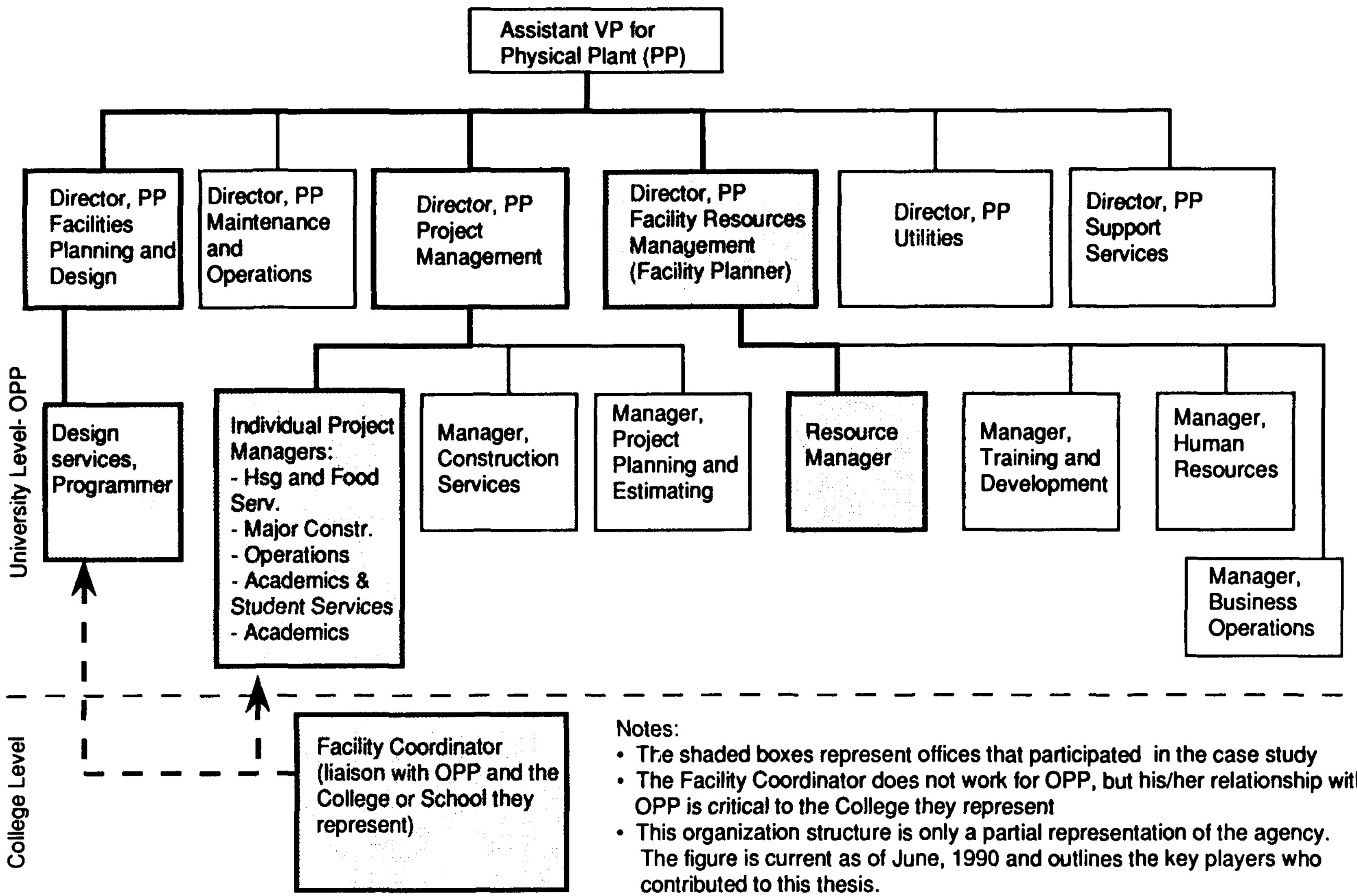
4.2. SCOPE

Projects were selected from a large university's main campus facility management program. The structure of the organization that manages the facilities is shown as Figure 4.1. In this thesis, this organization will be referred to as the Office of Physical Plant (OPP). OPP can support planning, programming, design, and construction/project management operations in-house, or can contract to have some or all of these services provided by outside AE firms. Funds to provide facilities (renovation, construction, leasing, etc.) come from various public sources.

A variety of projects were intentionally chosen. The characteristics that were varied include the project's complexity, building type, or method of programming. The nature of the renovation or construction effort was also different among the projects. For example, some projects required new construction, while others involved only renovations to existing facilities.

4.3. OBJECTIVES

There were two principal purposes when structuring the interview questions. The first was to verify **completeness**--that necessary information was included. In this case, the project programs were compared against the FPPM to test the adequacy of the program. The second objective was to identify **criticality**--what information was essential and why. Each OR was asked to summarize the critical elements of the program.



Notes:
- The shaded boxes represent offices that participated in the case study
- The Facility Coordinator does not work for OPP, but his/her relationship with OPP is critical to the College they represent
- This organization structure is only a partial representation of the agency. The figure is current as of June, 1990 and outlines the key players who contributed to this thesis.

Figure 4.1: Office of Physical Plant (OPP) Organization Structure

The results of these objectives are outlined in section 4.6 and more detailed information is presented in Appendix D The lessons learned from the case studies were structured as guidelines for the owner's representative. In order to understand the perspective each OR had, it is important to look at his/her role with respect to planning, programming or managing the projects. The OR's relationship to the project is discussed in section 4.5.

4.4. METHODOLOGY

This section identifies the method of gathering data during the case study. First, the project familiarization phase is discussed, then the interview process is outlined.

4.4.1. Familiarization

Data was collected through background investigation and interviews with project level owner's representatives. The first phase of the case study involved becoming familiar with the general scope of the projects. Project familiarization was accomplished by reviewing related feasibility studies, master plans, any available programming documentation, plans and specifications, and as-built information. After the background investigation was completed, interview questions were developed for the ORs associated with the various projects.

The project codes, phases, and basic background information is shown on Table 4.1. These are the project codes: planning (PL), design development (DD), construction (C), and post occupancy (PO). The general scope of each project is summarized below:

PLANNING

• PL 1: New technical research space; need 100,000 net assignable square feet (NASF)

• PL 2: Renovation of existing space, to be leased by University

DESIGN

• DD 1: Renovation of technical laboratory space--ventillation, laboratory tops, AC, etc. (NSF matching funds- which presented major project constraints)

• DD 2: New research, instructional, and feeding facility; 35,000 gross assignable square feet (GASF)

CONSTRUCTION

• C 1: Renovation of existing laboratory space to support new research programs

• C 2: New construction--5 story building with 28 classrooms and administrative office space; 90,000 GASF

POST OCCUPANCY

• PO 1: New construction--three story "generic" research laboratories to be used as temporary space; 33,000 GASF

• PO 2 : Renovation of previous kitchen space for office and administrative purposes; 1900 NASF (being leased by University)

Table 4.1: Summary of Case Study Projects by Phase

PHASE • PROJECT	# OF USING AGENCIES	COST	PROGRAM BY	DESIGN BY
PLANNING				
• PL 1	2	$ 10 Million	A/E *	A/E
• PL 2	6 (Leased)	$1.5 Million	I/H	A/E by owner
DESIGN				
• DD 1	1	$2.3 Million	I/H	A/E
• DD 2	1	$6 Million	A/E**	A/E
CONSTRUCTION				
• C 1	3	$6 Million	I/H	A/E
• C 2	2	$11.2 Million	A/E**	A/E
POST OCCUPANCY				
• PO 1	1	$3.4 Million	A/E**	A/E
• PO 2	1 (Leased)	$.3 Million	I/H	A/E by owner

Key:

* The program is not yet fully developed

** A "firm" (separate) program was not developed

I/H- In-House (work was done by the university's staff)

A/E- Architect/Engineer (work was done by contract)

# of using agencies- how many different organizations were using the space.

4.4.2. Interviews

Fifteen interviews were conducted with project level ORs. The different ORs are discussed in the next section. The ORs interviewed for the various projects are shown on Table 4.2, categorized by the role each played in the project. The ORs were interviewed to determine:

- What programming methodologies (gathering, storing and organizing information) were used?
- What programming information in the program was most critical to them?
- Was the program document used during the life cycle? If so , how?
- What were the strengths and weaknesses of the programming process and product?
- Could the facility have been better if program information was utilized during all phases of the life cycle?

The link between planning and design was emphasized, by conducting interviews with the full range of ORs for the design projects. A complete set of the questions used during the interviews is provided in Appendix D.

Table 4.2 Summary of Owner's Representatives Interviewed:

PHASE • PROJECT	Planner	Facility Coord.	Project Manager	Programmer
PLANNING				
• PL 1	*	●		**
• PL 2	●	●		●
DESIGN				
• DD 1	*	●	●	
• DD 2	●	● (user)	●	
CONSTRUCTION				
• C 1	●		●	
• C 2	*		●	
POST OCCUPANCY				
• PO 1			●	
• PO 2			●	

Notes:

● Indicates the OR was interviewed in this phase

* Indicates the FC acted as the planner for this project

** Indicates the program is not yet fully developed for this project

4.5. OWNERS' REPRESENTATIVES

Four different types of owner's representatives were interviewed during the case study. The next section identifies their titles and discusses their roles and responsibilities related to the project.

4.5.1. Facilities Planner/Resource Manager

The facilities planning and resource management (RM) staff ensures valid project requirements are represented to the decision making body (bodies) of the university for approval to fund construction/renovation. The RM staff supports the university's strategic goals regarding construction and renovation, as well as acting as the initial interface between OPP and the FC.

4.5.2. Project Manager

The project manager (PM) is assigned to a given project early in its life cycle. He/she works with the using agency to develop the initial program, then to coordinate project schedules and the selection of design services (if applicable). The PM also conducts design reviews and monitors construction progress.

4.5.3. Programmer

The programmer is the person who develops the program for the facility renovation or construction project. In-house programming us usually done by architects and/or engineers in the Design Services section. In the past, some programming was done by the facility coordinator.

4.5.4. Facility Coordinator

The facility coordinator (FC) is the College level representative for the "using agency" and acts as a liaison with OPP. The FC manages the college's facilities as well as minor and major construction programs. Although the FC's role varies slightly between various college's or schools, his/her responsibility is to ensure that individual departments or research centers have clearly communicated their requirements as users of the facility to OPP.

4.6. CASE STUDY FINDINGS

The findings are presented in two general categories: completeness and criticality (as they were defined previously in this chapter). General remarks and observations are presented in the summary which follows.

More detailed case study data is presented in Appendix D, along with a sample of the interview questions. The appendix also summarizes the results from the background investigation and interviews. The significant findings from the interviews follow.

4.6.1. Completeness (Based on the FPPM)

A summary of the information contained in the facility file, and/or the facility program for each project is shown as Table 4.3. Table 4.3 summarizes the completeness of the various programs and highlights the critical information. The table also separates hard and soft issues and provides general remarks applicable to each program.

Overall, none of the facility programs (as they were defined in the project records) contained all the information specified in the FPPM. However, the ORs interviewed agreed the information categories proposed in the FPPM should indeed be presented in a program.

4.6.2. Criticality

The information "most" critical to a project was typically specific to the goals/constraints of that project environment. However, budget was a critical issue in all but one of the projects. Ensuring functional issues were satisfactorily addressed was critical in all projects, especially in more technical facilities where performance criteria were essential (i.e., research laboratory space).

Typically, functional issues are identified in any program. Preprogram information was seldom referenced in the program. Social issues were typically identified, especially if involving unusual regulatory compliance requirements. References to the project schedule and budget were often left

Table 4.3: Case Study Program Completeness and Critical Information

Project	Pre-Program	Soft Issues:		Hard Issues			Remarks
		Social Issues	Form	Function	Econ	Schedule	
PL 1	1	1	0	**1** a	1	1	Initial program IH (for capital budget approval)
PL 2	0	1	1	**2** a	**1** b	**0** c	Method: IH. Utilized detailed technique for space analysis.
DD 1	0*	1*	0	**1** a	2*	2*	Initial program IH, then AE. NSF req. drove program (QL)
DD 2	0*	2*	2*	**2*** a	**2*** b	2*	No formal program. Req. Id'd by design review process.
C 1	0*	2	1	2	**2** b	**2** a	Method: AE. Excellent process for environmental req.
C 2	0*	2*	2*	**2*** a	2*	2*	AE--Initial program not revised after 2 major scope changes.
PO 1	0	0	1	**1** a	**2** b	**1** c	Method: AE. User input was limited
PO 2	1	2	2	**2** a	2*	2*	Key IH program--showed OPP the value of "space" program

Notes:
0- This information was not in tne program
1- Schematic level of information was evident in the program
2- Schematic and detailed level information was evident in the program
*- Information was in the facility file, but was not in the program (either there was no program, or no current program)
1 - Bold large letters indicate information critical to the owner's representative
Footnote references in the table indicate the hierarchy of critical information.
The key is: [a] Identified as the most critical element of the program
[b] Identified as the second most critical element of the program
[c] Identified as the third most critical element of the program

out of the program, but were a critical part of the facility file. Social issues were only critical when they involved sensitive political issues. If the politics surrounding a social issue affected approval from one of the decision making bodies, then it was critical. Time (the project schedule) was critical more often than not. The form of the facility, or of the spaces within a facility were usually a response to the function of the facility.

4.6.3. Summary of Other Interview Results

This section summarizes responses to some of the key questions posed during the interview process. One of the desirable characteristics of a program identified in this thesis is the ability to use the program as an evaluation tool during the life cycle of construction. The following tables summarize the results of interview answers related to evaluation. Table 4.4 tabulates the responses to the question "can the program be used to evaluate contractor performance and the performance of the completed facility?" Tables 4.5 and 4.6 provide examples of other program information useful to the OR in evaluating the construction and operations phases respectively.

The programs usefulness in evaluating the design product is easily understood. There is a direct relationship between the problems (requirements) identified by the program and their synthesis (solution) during the design process.

Table 4.4: Using the Program to Evaluate Construction and Operations of a Facility

Question:	yes	no
Can the program be used to evaluate contractor performance and the performance of the completed facility?	15	0
• Use during construction	15 (2 had a qualified "yes")	0
• Use during operations	15	0

Note: These were the qualified "yes" responses.

1. There is an indirect relationship between the program and construction--through the contract documents. Since the program was used to develop the design, then contract documents, there is a link between programming and construction.
2. The contract documents should be used. The program's design intent is useful, but could only be used informally.

Table 4.5: Program Information Useful to Construction Evaluation

What information was/is most useful to evaluate the construction of the facility?	Number of times this answer was given
• Building system performance criteria	7
• Quality	7
• Time	5
• Function	4

Note: Other suggestions included the installation of equipment, the number of change orders (checking to confirm design intent), size and cost.

Table 4.6: Program Information Useful to Operations Evaluation

What information is/was most useful to evaluate the operations of the facility?	Number of times this answer was given
• Function	9
• Building system performance criteria	6

Note: Other ideas mentioned by the ORs included facility utilization, size, utilities, quality, productivity, and performance of the research space. These are specific functional or system performance criteria issues--reinforcing the importance of that information.

4.7. CONCLUSION

The results presented above reinforce the information presented on Table 4.3, that the hard issues, function, economy and schedule, are critical elements of the program. Since building system performance criteria are considered functional requirements, function is the most important type of information when evaluating construction or operations phase performance.

The program needs to have a current information. Unfortunately, in two cases (projects DD 2 and C 2) a program was either not developed, or was not updated as major scope changes occurred, resulting in substantial project delays. This was seen as a mistake and the common consensus among ORs was that a program should be developed before design. The result--the university studied has recently made policy changes reflecting the lessons it learned. Currently, a "programming committee" is established at the onset of every project and initial in-house programming is conducted before outside design professionals begin either programming or design. The committee focuses on getting using agency input early in the programming process, then establishing and maintaining close communication between the various members of the programming team.

The FPPM was confirmed by the OR interviews as a valid framework for programming. Based on this case study, suggested guidelines for developing the program and evaluating criteria are presented in Chapter 5.

Chapter 5

FACILITY PROGRAMMING GUIDE

5.1. GENERAL DESCRIPTION

The Facility Programming Guide (guide) presented in this chapter is developed using the literature and lessons learned from the case study projects. The scope of the guide is consistent with the scope of this thesis. *In this chapter, the objectives of developing the guide are outlined,* the users of the guide are described, and starting in section 5.4, the guide itself is presented in its entirety. Standard forms for gathering information are also discussed and an example of using the guide is provided.

5.2. OBJECTIVES

The guide has two basic objectives. It can be used as a **checklist** for developing the program, or as a **training tool** for personnel not experienced in thorough or sound programming procedures. In either case, using the guide can provide a programming standard to an agency, improving the proc s and product of the program and the facility.

5.2.1. Checklist

The guide can be used in conjunction with the FPPM to lay the foundation for the format and procedures needed to develop the program. The checklist-like guide is in fact a set of guidelines that leads the user through the program. Emphasis is on the use of the process suggested by Peña to develop the program (1987, see Chapter 2). This method would then be used in conjunction with a system for evaluating programming information on a common basis. Evaluating information in the program will be most effective when following the guidelines included in section 5.4.1.2.

5.2.2. Training Tool

The *guide can be used in conjunction* with the FPPM to orient personnel in training to the guidelines suggested here. The guide lends itself to training because it prescribes standards for gathering information, analyzing that information, and presenting the program in a standard (but flexible) format. The evaluation process identified in the guide can also be a valuable way for new trainees to learn more about the organization's decision making process.

5.2.3. Users

The guide was developed for the public sector OR. In this context, OR refers to those representatives defined in Chapter 4. Other public sector agencies (i.e., state, federal, department of defense, etc.) may be able to adapt this guide to their programming process, because many funding issues, and even bureaucratic processes, are consistent among different public sector agencies.

Readers who are interested in adapting this guide to their specific programming and design environmen' should consider their agency/firm's specific requirements, limitations, approval mechanisms, etc. Unique office requirements will affect the overall **process** of programming a facility. However, the facility team should determine what specific information the program identifies. The **product** may vary from project to project, but the facility team should consider the information identified in the FPPM before deleting any categories of information.

The term "programming committee" used in Chapter 4 identified the people who are establishing the building requirements at the working level. The committee may be small, as two-three people (on a small scale, simple project), or large, say 25 people or more, (when working on a large, complex project--typically with complicated interface and approval processes). They must identify what information is critical to them (or their organization) and ensure the program is developed to support those information needs.

5.3. SCOPE

This guide was developed for the public sector owner's representatives (OR) in the large university environment. The guide is based on the lessons learned by this particular facility management agency (referred to as OPP). The lessons learned in the case study are identified in detail in the section 5.5.

5.4. THE FACILITY PROGRAMMING GUIDE

The guide establishes a foundation for developing the program. Specifically, it acts as a "road map" for the facility programmer by presenting a format consistent with the information framework of the FPPM. Background information regarding the programming process and evaluating information in a program is outlined for users of the guide. The format for the guide is presented, then guidelines for using standard data gathering forms are identified.

5.4.1. Using the Programming Guide

The guide is presented by first identifying programming process related issues. Some fundamental guidelines regarding the evaluation process are presented. These evaluation guidelines are generic in nature and are presented to reinforce how on might use the value cell of the FPPM. A checklist for using the guide is then presented. Lastly, a format is suggested for the guide and general rules of thumb are identified.

5.4.1.1. Process

The recommended generic process for programming resembles the scientific method and its embodiment as described by Peña (1987). Peña's adaptation of the scientific method for programming is shown here in Table 5.1.

Table 5.1: Peña's Five-Step Programming Process (1987)

Generic Programming process	Peña's five-step process
Gathering information:	1. Establish **Goals**: "what does the client want to achieve and why?"
	2. Collect **Facts**: "What is it all about?"
	3. Uncover **concepts**: "How does the client want to achieve the goals?"
Testing feasibility:	4. Determine **needs**: "How much money, space and quality?"
"Distilling what you've found":	5. State the **problem**: "What are the significant conditions and the general directions the design of the building should take?"

5.4.1.2. Evaluation

The importance of evaluating information in the program was identified by White (1972) and was confirmed through the case study (see Chapter 4). White discussed evaluation generically, stressing the importance of evaluating information in the program. The case study discussed the importance of using the program to evaluate the design, construction, or operation of a facility. Both types of evaluation should follow these general precepts:

- Evaluation is, by definition, a value judgement. The evaluation of elements in the program should be based on a consistent "valuation" of the different elements in similar terms. For example, hard issues (economy, function, and schedule) can be easily evaluated on similar terms. However, when hard issues are being compared to soft issues (i.e., what form the ' icility should take) they must be compared on similar terms.
- Understanding the generic control process is important toward understanding evaluation. The generic control process can be represented by this cyclic process:

  1. Set the standard
  2. Measure actual performance
  3. Compare performance to the standard
  4. Reset the standard or modify the performance methods

  This cycle is important because a "standard" or goal is required to effectively evaluate anything.
- Evaluation is a critical part of the decision making (DM) process. Simon (1960) identified a three part process:

  1. Intelligence- gathering information about the problem
  2. **Design**- Considering different options or alternatives
  3. Choice- Selecting one of the possible options

There are also two critical steps after a decision has been made. The choice must be **implemented** (executing the option), then **monitored** (tracking the result, providing feedback to the DM body). Evaluation occurs at the "design" and "monitor" stages of the process. The program can be a tool that supports decision making when the **priority** and **value** of the building requirements are used to compare program information.

- Be specific when identifying project goals or standards. According to White (Introduction to Architectural Programming, 1972), "the more declarative or specific the goals, the easier the task of evaluation".

5.4.2. Checklist for the Using the Programming Guide

This checklist incorporates Peña's five-step process for programming and emphasizes important events that may require the guide to be evaluated and/or updated. The checklist is presented as Figure 5.1. Figure 5.2 shows examples of various events that may cause the programming team to evaluate (or reevaluate) information in the program. The listing is not all inclusive, but does illustrate some key decision points.

5.4.2.1. Format for the Program

The format for the program is flexible. Different project have unique internal and external constraints which should be recognized. These constraints often define what information is critical to the project.

Facility Programming Guide Checklist

1. Establish **Goals**: "what does the client want to achieve and why?"
2. Collect **Facts**: Identify preprogram information that significantly affects or constrains the facility.
3. For each general category of information is the FPPM, identify the **critical issues** for this project (sort by category).
4. Review the specific categories of program information as a cross check to insure all relevant information is considered (Figures 3.3 and 3.4).
5. Review each item of program information considering each type of information identified by the FPPM address and utility codes. Complete the framework (of the FPPM).
6. Use the standard form for functional analysis (as appropriate) to develop the functional requirements of each activity/space.
7. Use the standard form for gathering data about equipment-specific requirements as appropriate.
8. **Update** the forms as required to maintain currency.
9. Continue to develop schematic information about the "problem." Do not develop design alternatives (solutions) until the problem is understood.
10. **Detail** the schematic information initially developed (annotate changes in "level" on the information framework).
11. Uncover **concepts**: "How does the client want to achieve the goals?"
12. Test feasibility--this is where the needs are determined--"How much money, space and quality?"
13. Clearly state the **problem related to each critical issue**: "What are the significant conditions and the general directions the design of the building should take?" Use the standard format , section 5.4.1.3)
14. Use the evaluation guidelines when studying "trade-offs" or analyzing information at key points in the construction life cycle. (Figure 5.2)

Figure 5.1: Checklist for Using the Guide

EVALUATING INFORMATION

The following example represents key points in the life cycle of the facility when the program should be reviewed and updated (as appropriate).

Manage/Plan

- Any time the scope changes
- Any changes in the facility team

Design

- When design development begins
- With any changes in scope
- To weigh design alternatives

Construction

- To review design intent related to building system performance criteria and the quality of construction

Operations

- To review design intent related to functional requirements (i.e., building system performance issues, the functionality of activities/spaces).

Figure 5.2: Checkpoints for Evaluating Information

Recognizing this, the final format of the program should be based on an agreement from the members of the "Programming Committee". The recommended format is presented as Figure 5.3. This figure shows an example of using standard word processing software to show the hierarchy of information in the program.

The program category "**Background**" should identify what the basis for the project is--the design intent. The section "**General Scope**" of the project should provide an overview of hard issues (function, cost and time) and soft issues (social issues and form). The next section, "**Summary of Critical Programming Issues**" should be a listing of all priority one and two information, sorted by category. Other information, called "**Remaining Programming Issues**" should be a listing of the priority three and four information. Preprogram information that is in the third and fourth levels of importance does not merit inclusion in the program.

Detailed information generated in the course of developing program information can be presented in either of two options: accompanying the information in the body of the report, or at the conclusion of the report. There are advantages to each method, but this author recommends including detailed data and table at the end of the program so the program can be presented as an executive summary, with the backup data being an "Appendix" to the report. However, graphic representations of ideas should be "liberally" included throughout the text to explain or reinforce concepts.

5.4.2.2. General Considerations

The general rules for using the guide are based on the lessons learned by OPP during their management of the projects in the case study.

1. BACKGROUND

2. GENERAL SCOPE OF THE PROJECT

- overview of hard issues: function, cost and time
- overview of soft issues: social issues and form

3. SUMMARY OF CRITICAL PROGRAMMING ISSUES

(this should be a listing of all priority 1 and 2 information, sorted by category vertically as well as horizontally)

3.1. Priority 1 information

- Function- Space analysis and building system performance criteria
- Economy- Overall budget
- Social Issues- For example, a sensitive political "agenda" item
- Preprogram information- Relationship to new research park
- Form- The image the new facility should present

3.1.1. Priority 2 information

- Function- Adjacency relationships
- Schedule-
- Social Issues

4. REMAINING PROGRAM INFORMATION

4.1. Priority 3 information

- Social Issues
- Form

5. APPENDICES AND SUPPORTING DATA

5.1. Life Cycle Cost Estimate
5.2. Detailed Space Analysis Calculations
5.3. College of Engineering Growth Projections

Note:
- In this sample, the details (data) are included at the end of the program.
- Priority level 4 information is not included in this example.

Figure 5.3: Suggested Format for the Program

Some of these rules seem "intuitively obvious". However, for one reason of another, mistakes were made during the course of the projects, resulting in the realization of the following "guidelines".

- Emphasize developing the program at the front end, before beginning design development. This needs to be adopted as the standard
- Assign a "program committee" as soon as the project is approved. Organize the programming committee (team) to make decisions at the lowest level possible (to "save time and aggravation")
- Actively get user input (and feedback) early in the process. Solicit end-user involvement (i.e. students) where it will benefit the project
- Institutionalize a **standard** system to be used by all the parties associated with the initial program development (see Figure 5.1 and the rules of thumb for using a standard form when space planning are shown in section 5.4.3)
- Develop standard equipment data sheets and use them, especially on "system intensive" projects. System intensive projects are those facilities whose function is critically dependant on the safe, efficient operation of a building system (i.e., laboratory, hospital, etc.). Take the time to teach the user how to use the data sheets
- When possible, the program should establish the budget, instead of the budget establishing the project scope.

5.4.3. Using Standard Analysis Worksheets

The following guidelines were developed based on a standard form for space planning currently used by Bob Myrick, in OPP's Design Services Section. The format for the space analysis sheet is presented, followed by a discussion of guidelines for collecting data.

5.4.3.1. The Standard Format for Functional Analysis

This section presents a standard format used for collecting information about how a using agency utilizes space for a given activity. It is important to think about "activities" when programming, as opposed to "rooms" in order to help the using agency think about how they use a space (and how they function in that space) generically. This format is standardized; however, the programmer should not hesitate to modify the form as needed. The format is presented as Figure 5.4. An AE firm (Elwood S. Tower Corp.) developed a standard form for equipment requirements on one of the case study projects. This form is shown as Figure 5.5 and is self explanatory.

5.4.3.2. Guidelines

The following guidelines (shown as Figure 5.6) are based on the interview the author conducted with Bob Myrick (the originator of the worksheet shown as Figure 5.4). These rules of thumb should be followed when developing the functional requirements.

1. (Reference #) The name of the activity/space

A. Space Purpose and Type of Activity:

B. Number of Occupants:

(both full time and part time)

C. Space Relationship:

D. Paper Flow Relation to Other Spaces:

(may be the same as space relationship, or may be different)

E. Workers' Foot Traffic Relationship:

F. Visual Relationship to Help Security and Control:

G. Office Type Furniture in Space:

H. Office Type Equipment in Space:

I. Other Equipment in Space:

J. Electrical Lighting:

K. Electrical Power:

L. Special Systems:

M. Ventilation:

(this should relate to equipment locations, etc.)

N. Plumbing Specialties:

O. Special Finishes or Space Needs:

P. Other Special Environmental Needs:

Note: the schematic layout (plan) derived from this worksheet uses the reference # to relate the space on the drawing to the worksheet.

Figure 5.4: Sample Space Analysis Form (Bob Myrick, OPP)

PROJECT: **Prepared by:** **Date:**

GENERAL

Item Description: ________________ Tag: __________ Manufacturer: ______________

Model/Catalog #: ________Options and accessories required: ______________________

Furnished by: O owner O contractor O other: __________

Installed by: O owner O contractor O other: __________

Quantity and locations: ______________________ Operating Schedule: _ ___________

Dimensions: ____________________ Weight: __________

HVAC

Heat rejection: ____________ BTUH

Exhaust rates: ____________ CFM Connection size: ______

Makeup air rates: ____________ CFM Connection size: ______

Filtration requirements: ______________ Control requirements: ____________________

PLUMBING

Cold water: _____GPM _______PSI size: _____ Temp: _____ oF

Hot water: _____GPM _______PSI size: _____ Temp: _____ oF

Steam: ______Lbs/Hr _______PSI size: _____ return: direct or floor

Compressed air: ______CFM ______PSI size: _____

Oxygen: ______CFM ______PSI size: _____

Nitrogen: ______CFM size: _____

Gas: ______CFM size: _____

Vacuum: ______CFM ______Hg size: _____

Other: ________________

Is distilled of deionized water required? O no O yes, GPM: ___

Is drainage required? O no O yes, GPM: ___

Is aciJ drainage required? O no O yes, GPM: ___

Is a vent required? O no O yes, GPM: ___

Is an emergency eyewash or shower reqd? O no O yes, GPM: ___

FIRE PROTECTION

Are there any special considerations for fire protection for this equipment? O yes O no

If yes, detail requirements: ______________________________________

ELECTRICAL

What kind of electrical connection is required?

O none required

O Direct, "hardwired"

O Plug?receptacle--NEMA type: ______________________

O Other: ______________________________

What is the required voltage? O 460 O 277 O 230 O 208 O 120

What is the phase configuration? O single phase O three-phase

What is the frequency? O 60 Hz O other: ______Hz

Is a neutral required? O No O Yes

Is an isolated ground wire required? O No O Yes

What is the full load in Watts: ________ or motor horsepower: _____

What is the full load in Amps: ________ or full load volt amps: _____

Is the load steady state or fluctuating? ________________

Does the equipment require surge protection? O no O yes

Does the equipment require line filters? O no O yes

Does the equipment require emergency power? O no O yes

Does the equipment require UPS? O no O yes

Does the equipment require voltage regulation? O no O yes If yes, Accuracy: ________

Is a disconnect switch required at the equipment? O no O yes If yes, O fused O non-fused

Figure 5.5: Sample Equipment Data Sheet (Tower Eng.)

Functional Analysis Worksheet Checklist

1. Try to "unclutter" the client's mind. Get them to focus on what is needed (functionally) for them to do their job best. Bob Myrick called this getting the client to "free think". For example, do not worry about cost when defining functional relationships, and support spaces.
2. The communication process is the key to gathering functional information. To facilitate this process, the programmer should following these steps at their initial interview with the client:
   a. Explain the purpose of the form
   b. Explain the form (walk-through the items on the form)
   c. Agree on a date for the first follow-on meeting, usually one week away.

   The client will fill the sheets out, then discuss the details with the programmer at the next meeting.
3. Get the sheets back. Input the data (should be on a word processor at least), then note questions related to specific information on the sheets. These questions should be clarified at the next meeting.
4. Set up a second meeting with the client. Schedule no more than four hours (or the meeting will be too long). Be sure to define terms so everyone has a common view point.
5. Go through the iteration of meeting with the client and discussing their functional requirements until both the programmer and the client are comfortable that the functional needs for a given activity are understood by both. (This may mean 4-5 iterations).
6. Develop schematics, based on the relationships of all the activities.
7. Develop a square footage summary of spaces, room by room. This is net assignable square footage (NASF); as a general rule, add 10-20 % to determine the gross assignable square footage (GASF).
8. Study ratios of walls to GASF, and circulation to GASF.
9. Look at basic square footage costs (based on historical data).
10. Identify the schedule (based on client needs and constraints).

Figure 5.6: Checklist for Using the Functional Analysis Worksheet

This process for developing functional requirements, when done "generically", creates a somewhat "timeless" program. For example, in one of the case study projects, seven different using agencies were selected to relocate to a facility. The facility needed minor renovation to support the new tenants. When the programming for all seven using agencies was completed, it was clear that only five of the seven could be accommodated feasibly in the available space. Fortunately, the remaining two space analyses (programs) could be readily updated when another location is found for those clients.

In another example, the functional program described above was used to modify a spatial layout during the construction phase of one of the case study projects. The program could also be used to review how well a facility is performing against the standards previously established. This type of evaluation could provide feedback critical to future renovations or documenting lessons learned from the project.

5.5. LESSONS LEARNED

The lessons learned from the case study presented here, form the basis for the guide. The general considerations shown in section 5.4.2.2. were a summary of the full set of lessons learned that follow. They were sorted into three categories. "General comments" do not refer either a positive or negative association with the experience for the OR. "Positive" comments reflect aspects of the program (or programming process) that went well. "Negative" lessons address those aspects of the program (or process itself) that should be corrected. The lessons learned are summarized on Table 5.2.

5.6. SUMMARY

The information framework presented in the FPPM was developed for the public sector user; however, both the guide and FPPM have applications in the private sector. Both tools identify the critical elements of the program. The main differences between public and private sector work are the potentially "radically" different economic issues affecting the projects. The social issues dealing with the projects impact on the local community often carry much more weight in the private sector, based on local zoning and building permit procedures, etc.

The guide has it's greatest potential when considered in the context of automated database systems that would allow the programmer to use the FPPM framework in an object oriented, or relational database file structure. Potential automated applications are presented in Chapter 6.

Table 5.2: Summary of Lessons Learned

	Lesson Learned:
General comments	• The sooner you start the program the better off you'll be • Institutionalize the system used for programming this project (a highly structure and thorough space planning study)
Positive:	• Emphasis on program development at the front end needs to be adopted as the standard. • The professional met customer requirements (user is getting what they want, even if it's not smooth) • It will be a good building when finished. We broke the initial mind set (addition to an existing building) which was a bad idea. • Overall, the equipment data sheet was a good tool (should modify the process though--i.e., provide better instruction, closer coord.) • Looked at ways to cut corners and save money , but still provide needed functions. • There is a value to student involvement (in planning committees) • Program enabled minimum design time (4 months instead of 5) and was developed in-house due to NSF funding constraints.
Negative:	• Knew the problems would be there at working level--but decisions were "politically motivated" • There was no written, defined program. The "inertia" of the job didn't allow any redirecting. Make a stronger statement to administrators-- program before design • Time--the program and design development needed more control (i.e., setting up tighter, definite milestones, then meeting them) • *You can probably never be* thorough enough (developing the program/reviewing documents...) • It is beneficial to "preprogram" (note- this refers to programming) or pre-plan. If this would have been done, we could have put the project to bed two years earlier. Spend the program time up front. • The "pre-plan" should establish the budget, instead of the budget establishing the project scope. • Do a program (there was no firm program here) • There should be more than one person developing the program, get more user input next time • *Additional rest rooms should nave been provided.* (the AE didn't think so, and PL should have "stuck to their guns") • Develop the program fully before "drawing lines on paper" • Budgeting and funding projects: we approach funding unrealistically; problems carrying money over from previous fiscal years

Chapter 6

CONCLUSIONS

6.1. OVERVIEW

This chapter compares the research results to the original objectives. The limitations of the FPPM and the guide are also discussed. Finally, areas of future research are presented.

6.2. COMPARING RESEARCH RESULTS WITH OBJECTIVES

This section presents each of the original objectives of this thesis, and the degree to which the research results satisfied each objective. The objectives were to define programming, develop the FPPM, test the FPPM, and develop a programming guide. Each objective is discussed below.

6.2.1. Programming Defined

The first objective was accomplished by reviewing the current literature on programming. Subsequently, the preliminary definition of programming was updated after conducting a review of 15 facility programs, interviewing three industry experts, and completing a case study of public sector projects. It was modified slightly to emphasize the role the program should play in the evaluation and decision making process. The new definition of programming is:

> *Facility Programming is the process of analyzing the owner's desires, needs, goals and objectives in order to define essential facility requirements and present those criteria to the designer. The program must establish and maintain an information framework which can be utilized as an evaluation and decision making tool throughout the life cycle of construction.*

6.2.2. Facility Programming Product Model (FPPM)

The second objective, developing the FPPM, was accomplished. The model was refined after reviewing 15 facility programs, and conducting interviews with industry experts. These refinements were informally validated by the owner's representatives when conducting interviews for the case study.

The FPPM creates a flexible framework. Industry experts identified the critical importance of function, schedule, cost, social issues and previously developed strategic plans. The priority of each is important for the programming team to understand as they develop the program and evaluate information in the decision making process. Evaluating program information, especially the evaluation of soft issues on similar terms as hard issues, was also shown to be significant.

6.2.3. Test of FPPM

The FPPM was tested in a case study of public sector projects on a major university's main campus. The test determined what information was critical to the owner's representative. That the framework needed to be flexible, was recognized after the case study showed the priority of information can (and often does) vary from project to project. This is a function of internal factors within the team (i.e., their individual goals and objectives) or external factors (i.e., the budget approved by the board of trustees). Another aspect of the model was confirmed to be critical to a program--the value. Every OR concurred with the importance of evaluating hard and soft issues on common terms. They also confirmed that cost is usually the common basis for that evaluation.

6.2.4. Programming Guide

The development of the programming guide as the last objective was also accomplished. The lessons learned from the case study were incorporated in the programming guide. The guide was written for the OR's view point, and could be used by planners, programmers, project managers or the facility coordinator in a variety of public sector agencies.

6.3. LIMITATIONS

Both the FPPM and guide are limited in their current application to the industry. The framework has been developed in this research; now each should be implemented in industry to test their ease of use and overall utility.

6.3.1. FPPM

The FPPM is designed to accommodate a data file structure. The framework organizes program information, but is not currently developed for immediate computer implementation. Although the framework is valid as a manual system, further development is needed to reach its full potential. The model could become a multi-media database (i.e., cost information developed on spread sheets, graphic information developed by a Computer Aided Design (CAD) system, word processes textural information, etc.).

The model was developed based on public sector constraints. Industry experts reviewing the model had extensive private sector experience, and incorporated some of those professional lessons into their review of the FPPM. Private sector constraints and professional practices vary from the public sector. As a result, private sector case studies using the FPPM could broaden its application to the construction industry as a whole.

6.3.2. Guide

The guide was also limited in the scope of its application. Its use in the public sector is valid, but should be extended to the private sector for the same reasons identified above.

The guide would also be more effective as a management tool if it had the ability to use an automated FPPM. Automation allows the user to process more information and output it in different forms. Therefore, reports for decision-making bodies could be more effectively managed. Additional computer issues are identified in the next section.

6.4. AREAS OF FUTURE RESEARCH

This section identifies future research possibilities related to computer automation. Further studies related to developing planning information and a system to improve facilities management are also presented.

6.4.1. Automation

Computer automation should be pursued as a way to enhance the programmer's ability to manage the data in a program. The FPPM is designed to support an "open architecture"--a systematic way to organize complex information. As such, current hypermedia applications like "Hypercard" or "Supercard" could be designed to manage the information in the framework. Other database management systems, i.e., relational databases or object oriented databases could also be developed.

6.4.2. Facility Planning

Programming is one of many processes within the planning phase of a project. The relationship of other aspects of the planning phase to programming should be studied. For instance, how does the site selection process, or the selection of a project delivery system affect the programming process? Another question would be, what is the role of the planning process and products in relation to design and subsequent phases of the work? Studying these questions can show the value of properly selecting the facility team, creating a contract structure that meets the needs of the members of that team, choosing the most effective site for the facility, etc.

6.4.3. Facility Management

Facility Management is a critical issue for owners with large facility complexes and utility infrastructures. These large corporate owners want a process to manage facility construction projects from cradle to grave (planning to operations). The FPPM creates a database of design intent which can be kept alive during this life cycle.

Research conducted by Beckett (1991) in the Computer Integrated Construction Lab at the Pennsylvania State University outlined an information framework for facility operators. Perhaps the integration of programming information in a framework with the information needed by facility operators would create the basis for a facility manager's information framework. This framework could be used to collect information unique to the facility manager (FM). The FM needs to document and track planning, design, construction and operations information related to the facility. A sample view of this synthesis is shown as Figure 6-1. (Note that the reader is referred to "An Information Framework for Facility Operators" by Beckett, 1991, for a detailed description of the FOIF model.)

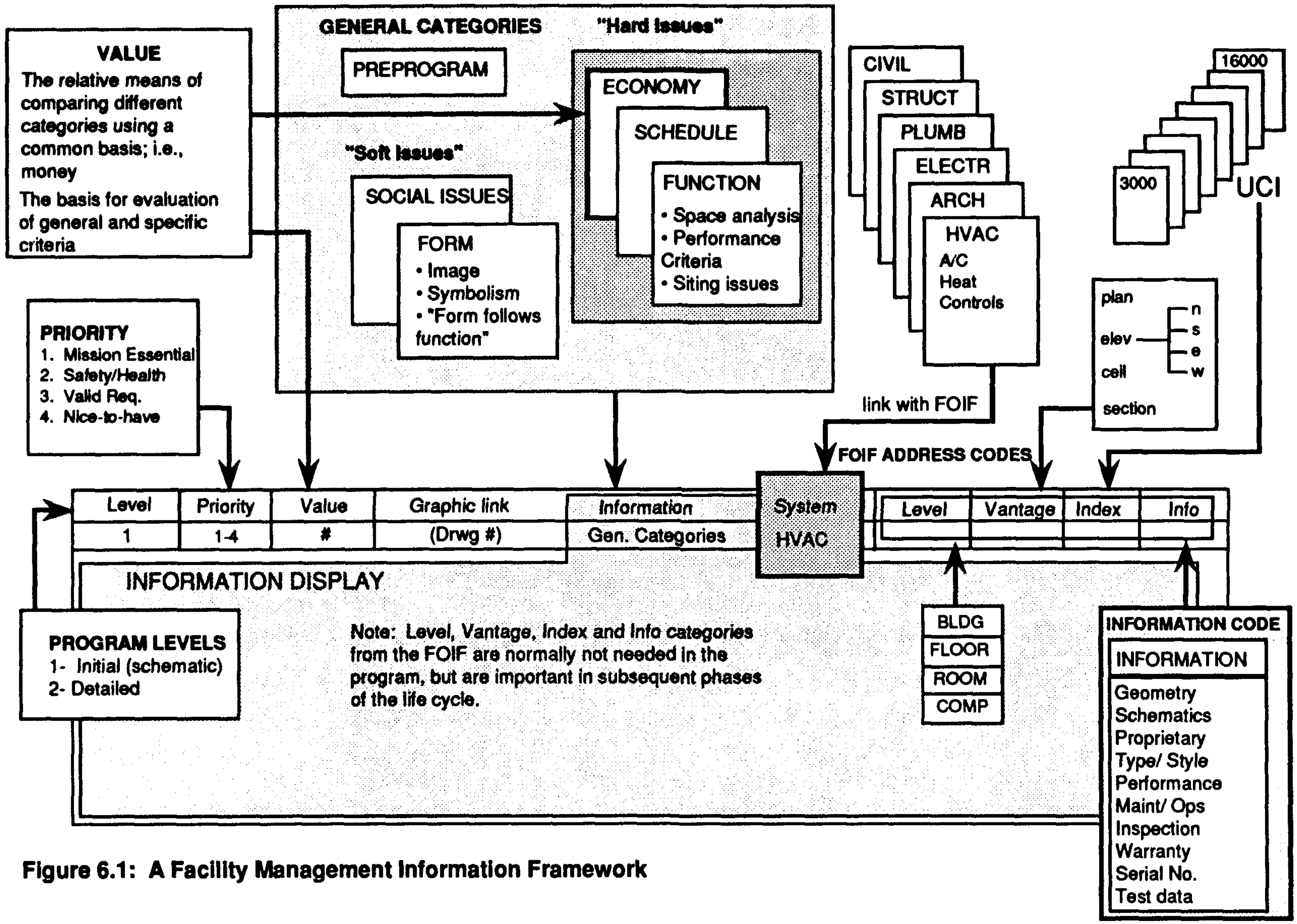


Figure 6.1: A Facility Management Information Framework

The FOIF contains address codes (system, level, vantage, and index) and an information code. The address codes serve as a way to locate information, whereas the information code refers to a specific type of information needed by a facility operator. The common "cell" between the FPPM and FOIF is the system code, which could serve as the link between the building requirements found in the FPPM and the detailed information needs of the facility operator.

6.5. SUMMARY

This chapter outlined the original objectives and the results of the research related to each objective. Limitations of the model and the guide were presented. These were based primarily on the narrow, public sector, applications of the research. Different programmers have their own value systems and professional experiences that affect their programming process. Areas of future research were discussed, noting the potential computer automation applications.

Overall, the FPPM serves as a way to organize and categorize programming information. The FPPM is based on the product needed to begin the design phase. Applying the rules for using the model in conjunction with the rules for using the guide can result in a standardized approach to identifying, analyzing, evaluating and presenting the building requirements.

BIBLIOGRAPHY

American Institute of Architects (AIA), DOCUMENT B141, "Owner-Architect Agreement," 14th ed., Washington, DC: AIA Press, 1987, pg. 5.

American Institute of Architects (AIA), Handbook of Professional Practice, vol. 1-3, Washington, DC: AIA Press, 1984.

Beckett, James P., "An Information Framework for Facility Operators," Computer Integrated Construction, Technical Report No. 23, Penn State, 1991.

Boyd, M.D. and James, E.M., "A conceptual Framework for Post-Occupancy Evaluation and an Analysis of Eleven Studies," Architectural Science Review, vol. 31, 1988, pp. 133-143.

Daish, John et al., "Post Occupancy Evaluation in New Zealand," Design Studies, vol. 3, no. 2, April 1982.

Davis, Gerald, "A Process for Adapting Existing Buildings for New Office Uses," taken from Facility Programming, Stroudsburg: Dowden, Hutchinson, and Ross Inc., 1978.

Evans, Benjamin H. and Wheeler, C. Herbert, Emerging Techniques - 2, Washington, DC: AIA Press, 1969.

Gordon, Douglas E. and Stubbs, M. Stephanie, "Programming," Architecture, May 1988, pp. 203-210.

Khayyal, Sari A, "Towards a Master Builder Information Framework for Project Developers," Computer Integrated Construction, Technical Report No. 16, Penn State, 1990.

Palmer, Mickey A., The Architect's Guide to Facility Programming, Washington DC: AIA Press and New York: Architectural Record Books, 1981.

Peña, William M. and Focke, John W., "Performance Requirements of Buildings and the Whole Problem," National Bureau of Standards, Special Publication, vol. 361, no.1, 1972, pp. 43-55.

Peña, William M., "Programming Management and Techniques," Conference notes from a national colloquy: Emerging Techniques of Practice Management, Penn State, 1969.

Peña, William, et al., Problem Seeking- An Architectural Programming Primer, 3rd ed., Washington, DC: AIA Press, 1987.

Perkinson, Gregory M., "A Summary of Program Evaluations," Working Paper, Department of Architectural Engineering, Penn State, May, 1991.

Preiser, Wolfgang F. E., Facility Programming, Stroudsburg: Dowden, Hutchinson, and Ross Inc., 1978.

Preiser, Wolfgang F. E., Programming the Built Environment, New York: Van Nostrand Reinhold Company, 1985.

Preiser, Wolfgang F. E., "The habitability framework: a conceptual approach towards linking human behavior and physical environment," Design Studies, Butterworth and Co., vol. 4, no. 2, April, 1983.

Royal Architectural Institute of Canada, Canadian Handbook of Practice for Architects, vol. 1, "Requirements and Predesign," March, 1977.

Royal Institute of British Architects, "The Project (Planning)-2, The Techniques and Methods," vol. 4, part 3 (3.500), July 1965.

Sanoff, Henry, Methods of Architectural Programming, Stroudsburg: Dowden, Hutchinson, and Ross Inc. (definition by the community development series editor, Richard Dopler), 1977.

Sanvido, Victor E., "An Integrated Building Process Model," Computer Integrated Construction, Penn State, 1990a.

Sanvido, Victor E., "Critical Project Success Factors (CPSF) for Buildings," Computer Integrated Construction, Technical Report No. 22, Penn State, 1990b.

Sanvido, Victor E., "Towards a process based information architecture for construction," Civil Engineering Systems, E. and F. N. Spoon Ltd., 1990c.

Simon, H. A., The New Science of Management Decision Systems, New York: Harper and Row, 1960.

White, Edward T., "Building Evaluation in Professional Practice," Florida A & M University, undated (a).

White, Edward T., Introduction to Architectural Programming, University of Arizona, Tuscon: Architectural Media Ltd., 1972.

White, Edward T., "Learning Decision-making for the Building Process," Florida A & M University, undated (b).

White, Edward T., Space Adjacency Analysis, Tuscon: Architectural Media Ltd., 1986.

White, Edward T., "The Value of Post Occupancy Evaluation to the Architect in Government," Florida A & M University, undated (c).

White, Edward T., "Programming Post Occupancy Evaluation and the Financial Success of the Architect," Tuscon: Architectural Media Limited, 1988.

White, Edward T., "Post Occupancy Evaluation from the Client's Perspective," Tuscon: Architectural Media Limited, 1989.

Appendix A

GLOSSARY OF TERMS

A.1 PROGRAMMING DEFINITIONS

The following terms are defined here to establish a point of reference for their use in this thesis.

Coded words: words assigned to arbitrary meanings. (Peña, 1987)

Economy: The efficient and sparing use of the means available for the end proposed. Implies as interest in achieving maximum results from the initial budget and the maximum cost/effectiveness of the operation and life cycle costs. (Peña, 1987)

Form: in design, form means the shape and structure of a building as distinguished from its materials. It is what you see and feel. In programming, form refers to what you will see and feel, avoiding the suggestion of a design solution. It's the "what is there now" and "what will be there." (Peña, 1987)

Framework: An open work frame. A frame of reference. A systematic set of relationships. (Peña, 1987)

Function: how the design product will work to do the job it is supposed to do. The performance. The "do"--the way people and things will move about to do the tasks they have to do. (Peña, 1987)

Functional requirements: Those requirements dealing chiefly with the way people will use the project (space) with convenience, efficiency and effectiveness. These, also, will involve the adequacy, the quality and the organization of space. (Peña, 1987)

Goal: The end toward which effort is directed. Suggests something attained only by prolonged effort. Goals can be classified as (1) project goals, and

(2) operational goals. Project goals are concerned with product; operational goals are concerned with process. (Peña, 1987)

Human factors: The programming considerations best characterized as having their basis for inclusion in the program due to either a behavioral aspect of the facility (behavioral factor) or the value system of one of the members of the facility team.

Human requirements: Those requirements stemming from the generalized human needs in terms of the physical, social and psychological environment to be provided. These human needs involve such general categories as self-preservation, physical comfort, self-image and social affiliation--initially expressed as specific goals. (Peña, 1987)

Information: Knowledge obtained from investigation, study or instruction.

Needs: Requirements; something necessary; an indispensable or essential thing or quality. (Peña, 1987)

Performance requirements: Those requirements stemming from the unique user needs in terms of the physical, social and psychological environment to be provided. These will involve the adequacy, the quality and the organization of space. (Peña, 1987)

Preprogram information: Information relevant to the programming of a facility that was assembled **before** the program was initiated, but is instrumental supporting or clarifying elements of the program. Examples are: feasibility studies, site selection studies, master plans, etc.

Program of requirements: Describes the document resulting from the preparation of the architectural program. (synonym--Design Brief, RAIC, 1977)

Requirements stage: Where the client identifies a potential project, collects pertinent data, prepares his program of requirements, and select the Architect. (RAIC, 1977)

Requirements: Something wanted or needed. (Peña, 1987)

Space requirement: Detailed listing of the amounts of each type of space designated for a specified purpose. (Peña, 1987)

Time ("Schedule" in the FPPM): Deals with the influence of history, the inevitability of change from the present and with projections into the future. (Peña, 1987)

Values: As they relate to architecture, are categorized as being one of three types: "enduring values" (firmness, commodity, and delight), "institutional values" (continuity or history of development), and circumstantial values (i.e., environmental--site, climate, urban context, etc., societal--cultural, legal, community, etc.)

Wants: Something lacking and desired or wished for.

A.2 PRODUCT MODELLING DEFINITIONS

Building levels: Hierarchically define and decompose a building into the different architectural levels of abstraction of a building (i.e., floors, rooms, components). These building levels are used to integrate the various discipline views of a building. (Khayyal, 1989)

Decomposition: To separate or resolve into constituent parts or elements, or into simpler compounds (Webster, 1954)

Discipline breakdown: Identifies the technical disciplines AEC (based on current practice) as being : Architectural, Civil, HVAC, Plumbing, Electrical, and Structural.

Model: "A purposeful representation"; a model doesn't have to be a complete representation; models have variables (factors that actively change) and parameters (factors that mediate the effect of the variables). Models can be descriptive or predictive and fit into one of three classes (Starfield, 1990):

- Mathematical (description of symbols for which we have a defined meaning; i.e., a structural calculation)
- Physical (visible geometric equivalents; i.e., a scaled 3-D model)
- Schematic (i.e., a flow chart)

White defines models as a way to understand information or operations and their relationships ...and serve as a means for organizing and presenting ideas about both ("Introduction to Architectural Programming", undated).

Product: The physical building related "non-informational" outputs of specific functions; i.e., the program is a product of the planning phase of a project, then becomes the input to the design phase of the project. (From Sanvido, "Towards A Process Based Information Architecture for Construction", (undated))

System: The term refers to the primary systems within the facility: architectural, civil, mechanical, electrical and structural.

A.3 INTEGRATED BUILDING PROCESS

MODEL DEFINITIONS (Sanvido, 1990a)

Provide Facility: Encompasses all activities required to provide the facility, from the initial establishment of need through planning , design, construction and operation.

Manage Facility: Includes all the business functions and management processes required to support the provision of the facility form planning through operations.

Plan Facility: Encompasses all the functions required to define the owner's needs and the methods to achieve these.

Design Facility: Comprises all the functions required to define and communicate the owner's needs to the builder.

Construct Facility: Includes all functions required to assemble a facility so that it can be operated.

Operate Facility: Comprises all of the activities which are required to provide the user with an operational facility.

Facility Team: Assembled by the owner to provide the facility. This team starts with the facility champion and expands to include representatives of the planner, designer, constructor, owner, operator, consultants, and facilities managers.

Appendix B

PROGRAMMING DEFINITIONS

PROGRAMMING DEFINITIONS

The following programming definitions were taken from various authors and show diverse frames of reference:

- "process by which criteria are developed for the design of a space, building, facility, physical environment, and/or any unit of the environment" (Evans and Wheeler, 1969)

- "the information and process that links those who design and build and those who use the resulting facility" (Dopler, from Facility Programming, 1978)

- "the process that elicits and systematically translates the mission and objectives of an organization, group, or individual person into activity-personnel-equipment relationships, thereby resulting in the functional program." (Preiser, 1978)

- "the *process* of determining what is needed by its (the new or existing building's) users and by the others who are affected by it (such as owners, managers and the public). Programming includes evaluating how the building satisfies these needs after it is occupied." (Davis, 1978)

- "...a way of systematically defining, ordering, and specifying goals, objectives, design intentions." (Dopler, 1977)

- "a process of identifying and defining the needs of a project and communicating the needs of the client to the designer." (Palmer, 1981)

- "the program document itself should be a comprehensive report that presents in text and tabular form the detailed quantitative and qualitative requirements of the entire client organization. The recommendations should include functional space standards, department -by-department space analysis and suggested organizational groupings which respond to adjacency, work and traffic flow requirements..." (Agostini, 1968)

• "The program is a document, the final output of the investigation phase of the design process. Its purpose is to predict those environmental conditions that are supportive and responsive to user's activity patterns. To be relevant, these predictions are constrained by an economic framework that is related to the construction process, the resources of the client and the time constraints of the project..." (Brill, from Architects' Guide to Facility Programming, 1981)

• "The building program is the central organizing force of the building; and, since a building is the crystallization of the social organization it contains, the building program must be the simultaneous specification of the organization and of the spatial relationships which are needed to house it..." (Davis, from Architects' Guide to Facility Programming, 1981)

• "...involves the unprejudiced analysis of a specific problem and its context. Because of its structure and reliance on techniques of interview and analysis and presentation in written rather than graphic form, programming remains the best time for analysis and clarity. It is usually the only phase of design during which the architect, user and owner can be compelled to explore and record their own prejudices and analyses of the solutions of others." (McLaughlin, from Architects' Guide to Facility Programming, 1981)

• "the process by which criteria are developed for the design of a space...and/or any unit of the environment. It is the means through which data about the needs of the ultimate building user are determined and expressed for the instruction of the Architect in the development of a design solution." (RAIC, 1977)

• Facility programming is the process of analyzing the owner's desires, needs, goals and objectives in order to define essential facility requirements and present that criteria to the designer. The program must establish and maintain an information framework which can be utilized as an evaluation and decision making tool throughout the life cycle of construction." (Perkinson, 1991)

Appendix C

EXAMPLE INDUSTRY REVIEW OF FPPM

FPPM Review Questions, William P. Ross, AIA

The following set of questions and answers is a sample of the industry review process. The same questions were asked to each of the three design/programming professionals. This represents the answers given by William P. Ross, AIA. Information and feedback that was outside the scope of the standard question format appears at the end.

C.1 BACKGROUND

1. Process: Who normally provides the program?

There is usually no program. In many cases, the program (as a document) is the initial set of schematics that the owner signs to indicate he concurs with the design intent (this is similar to the problem definition).

2. Product: What format does the program usually take?

The crudest version of the program is the first set of drawings.

- *How is it organized?*

This varies depending on the job. It is based on the:

- size of the job (cost)
- complexity
- the dictates of the job/client (i.e., need for approvals, feedback required, etc).

3. What type of information is normally in a program?

The information is normally a combination of hard and soft information:
Hard information:

- Master plan, strategic plan (including this information is critical, but only if the strategic information is well thought out..consider this from day 1)
- Function (the fu ional relationships)

- Where do I spend the money? (this is the trade-offs we face when looking at how the available funds are distributed throughout the project...what elements have priority, etc)

Soft information:

- Values
- Human factors
  - *How would you organize it?*

Discussed above, 1st set of drawings

4. What hierarchy does the info take (what is critical/essential)?

A system of **evaluation** must be added to the model to "value" the information. The client and Architect determine the value.

Focus: tie the program to the strategic plan(s). How does the parent corporation view this? Note: individual('s) goals can not be allowed to conflict with the strategic goals.

C.2 PERCEPTIONS OF THE FPPM

5. Do you agree with the necessity of the various information categories shown in the FPPM framework?

The categories are alright. These notes relate to different categories:

- Economy--relate this to the holding period (key to other objectives...)
- "Form results from the program". He was relating form to aesthetics and views it as being not related to function. Aesthetics can be a criteria, example--"Site's" work of the Best Warehouse. Question: what is the visual appeal worth?
- Time:
  - irrelevant when related to historical events
  - function of exogenous factors; i.e., operating/opportunity costs,
  - clarify this topic
- Values: measure in economic terms.

6. Are there other categories of programming information that should be included? If so, what are they?

A place should be added to the framework to show the value of the information. This relates to developing a system to evaluate the information in the various programming categories.

7. What type of programming information is most critical? How would you prioritize the categories?

The critical tool in the owner/architect relationship is managing the client. The public sector can learn from the private sector in this regard.

8. What factors affect that priority (i.e. owners values, programmer's experience (or bias)?

Evaluation: Evaluating the worth of the various categories is the essence (or should be the purpose) of the FPPM. Dollars and cents should be the common "point" for evaluation in the data base. You should be able to evaluate the factors in each "line" of the model (categories). Key--the criteria should be measured against something. When you put the information in, then get it back out, how do you measure it in a meaningful way? (apples to apples)

9. Does the program have applications throughout the life cycle of construction?

Yes. Look at FM (facilities management) as a source of input (and as a new market for architects in the future. This could be a database the Architect develops for subsequent FM) Large banks and property developers are working on this now. The program and FM should keep the Architect in the loop. You could view this as cradle to grave PM (Project Management)

10. What information might be useful in each of those later stages?
   - *Construction Phase*

Note: the program is a performance spec. This relates to construction.

- *Operation Phase*

See the discussion of FM above.

C.3 PERCEPTIONS OF THE INDUSTRY

11. What are the AIA standards for programming?

There are no clear standards

12. Are these guidelines enough?

No (because there are no guidelines)

13. What should be done to further the industry?

- *By professional societies (AIA)?*

The AIA should do more to improve professional development

- *By higher education?*

Yes

- *By the firm and the individual (continuing education)?*

Yes

C.4 MISCELLANEOUS NOTES

- Study private sector work. There are more factors that affect the program, and the work would be more exciting
- How do you evaluate hard and soft factors?
- How do you differentiate between exogenous (external) and indigenous (internal)? How do these factors affect decisions?
- The program is a performance specification, which is translated into an objective specification (design)
- How do you define the **actual** versus **perceived** value?

Appendix D

CASE STUDY INTERVIEW QUESTIONS

AND

CRITICAL PROGRAMMING INFORMATION

D.1 INTRODUCTION

This Appendix presents the questions asked to the owner's representatives (OR) during the case study of eight public sector projects. Then, information critical to the OR is presented in table form. The reader is referred to Chapter 4 for a more detailed description of the scope and objectives of the case study.

D.2 INTERVIEW QUESTIONS

These are the interview questions asked to the various OR:

1.1. PROJECT:

(The project name was entered here)

Owner's Representative:
(name)

1.2. GENERAL BACKGROUND INFORMATION:

As the owner's representative, what was your principal objective when developing the program?

Did anything happen in the early stages of project planning that significantly affected (positively or negatively) the program?

1.3. PROGRAM REQUIREMENTS:

How did you identify these issues:

- cost/financing

- function (of spaces in the facility)

- project schedule

- social concerns (facility image,

What was the format for each?

What information was left out of the program that should have been included?

What program information may be useful to you beyond the design phase (i.e., construction or operation)?

1.4. PROGRAMMING CATEGORIES OF INFORMATION

1.4.1. "Soft Issues"

1.4.1.1. Preprogram

How did you relate this project to "strategic information" (i.e., to support or justify the project)?

<u>Plan</u>	<u>Role</u>
Master Plan	
Proposed Capital Budget Request	
Space and Facilities Plan	
College (level) Strategic Plan	

How was it referenced in the program?

1.4.1.2. Social Issues

Were there any significant social issues affecting the project?

What were they?

How did (should) they have affected the work?

What unique regulatory requirements affected the work?

How did they affect the program (project)?

Did the program address any issues related to human behavior?

- How?

Did the value system(s) of the institution (college, or department) affect the decision making process related to the program?

- How?

1.4.1.3. Form

What factors affected (or should have affected) the form of the facility? consider:

- "form follows function"-
- symbolism (image)-
- based on performance-

How were these "form giving" considerations identified?

1.4.2. "Hard Issues"

1.4.2.1. Function

How were functional issues identified and presented?

- From your perspective, how effective was this process?

- If not well done, how could it be done better?

Performance Criteria

What building system performance issues were stipulated in the program (i.e., lighting levels, acoustic properties, ventilation req. etc.)?

What other factors should have been included?

Is there a standard, or is this project specific?

Relationship to the site:

What affect did the site have on the function of the facility?

What affect did the function of the spaces have on:

- the selection of a site?

- placement of the building on the site?

How did parking, vehicular and pedestrian movement impact the planning and development of the site?

1.4.2.2. Time

How were the design/construction time lines originally developed?

- Did the milestones change often?

- What was the impact of changes to overall project milestones

What constraints affected (determined) the project schedule?
Examples:

- Department/college goals-
- Funding constraints-

• Research milestones-

1.4.2.3. Economy

What is the project budget? (How many times did it change?)

What is the construction budget?

What is the source of funding?

What funding constraints (regulations) affected the project?

Were "hard" issues like cost used to evaluate other "soft" issues (like what image the building should present)?

• Could they have been?

1.5. SUMMARY

Can you use the program to objectively evaluate contractor performance? What information would be the most important in doing this?

Can you use the program to objectively evaluate the performance of the facility? What information would be the most important?

Did the program assist the decision making process (whether in design, construction, or operations)?

Summarize the critical elements of the program.

What lessons did you learn from the program's successes and failures?

D.3 CRITICAL INFORMATION

The critical information to the OR was determined from three of the questions listed above:

1. As the owner's representative, what was your principal objective when developing the program?
2. Summarize the critical elements of the program.
3. What lessons did you learn from the program's successes and failures?

The answers to these questions are shown by project on Tables D.1-6. Answers to all the questions are currently kept on file by the author. Finding out how the OR viewed their role in the programming process was instrumental when developing the guide (see Chapter 5).

D.4 CONCLUSIONS

As the reader can tell from looking at Tables D.1-6, the OR's critical information often varied from project to project. It is also clear that the presence of a well defined program contributed to the OR's view of success (interpreted through conversations with them, and the positive lessons learned on those projects). For example, project PO 2 utilized a detailed analysis of the functional activities in the program. The using agency went from 2300 NASF to 1900 NASF; and was still satisfied with the quality and utilization of their new space. In another example, project DD 2, the original program wasn't updated to coincide with major changes in the scope of the work. As a result, the design process was very "painful" to the OR involved. The concluding thought is intuitively obvious, but should be stated none the less--define, develop, evaluate and update program requirements, then enjoy the benefits during the design, construction and operation of the facility!

Table D.1: Planning Projects PL1 and PL2--Critical Program Information

PROJECT	O. REP'S PRIMARY OBJECTIVE	CRITICAL PROGRAM ELEMENTS	LESSONS LEARNED
• PL 1 (Eng/ EMS) PL--	• Provide necessary space for research project expansion	1. *Functional utilization* 2. Interaction between activities 3. Best "bang for the buck"--Ratio of NASF to GASF.	• The sooner you start the program the better off you'll be
• PL 2 (Rider 2) PL--	• Time frame (crunched for space, originally, needed to provide *relief to another facility*)	1. The "base program" for Fit--does the function for all the users work? 2. Systems 3. Function of a given activity	• <u>Positive:</u> Emphasis on program development at the front end needs to be adopted as the std. • <u>Negative:</u> Knew the problems would be there at working level--but decisions were "politically motivated"
FC--	• Find a location for two new programs	1. Time (because we're behind) 2. money 3. Function--critical to directors	• To early to tell
PR--	• Resolving new functional issues and correcting existing code deficiencies.	1. The non-written part: the *communication facilitated through* the program. 2. Does the client understand what is happening?	• <u>Positive</u>: We were able to save the programs *for two of the users who "bumped" from the* building <u>Negative</u>: The administration tried to put too much in the building (initially) • The building itself is inefficient (due to the original program and design)

Table D.2: Design Project DD1--Critical Program Information

PROJECT	OWNER'S REPRESENTATIVE'S PRIMARY OBJECTIVE	CRITICAL PROGRAM ELEMENTS	LESSONS LEARNED
• DD 1 (Fenske Lab) PM--	• Upgrade facilities as much as possible	1. Upgrade environmental systems - AC - Ventilation - Clearstory glazing	• Program enabled minimum design time (4 mo.s instead of 5) -- program was developed I/H due to NSF funding constraints.
PL--	• Upgrade outdated physical facility for a department	1. Upgrade Mech 2. Update the aesthetics of the labs	• To early to tell

Table D.3: Design Project DD2–Critical Program Information

PROJECT	OWNER'S REPRESENTATIVE'S PRIMARY OBJECTIVE	CRITICAL PROGRAM ELEMENTS	LESSONS LEARNED
• DD 2 (HRIM) PM--	• Meet the customer's needs • Identify requirements from other team members: - Reviewers (tech) - Resource Mgmt ($$) - User/College - Maint. and Ops (at OPP)	1. Functional Requirements: - Dining area (research) - Kitchen area (research) - HVAC research program	• Negative: There was no written, defined program - The "inertia" of the job didn't allow any redirecting - Make a stronger statement to administrators--program before design • Positive: The professional met customer requirements (user is getting what they want, even if it's not smooth.
PL--	• Get the user in a new location with adequate space	1. Cost hamstrung the committee 2. "research" bldg--key on it being "flexible, different"	• Negative: Spend program time up front - (here, spent to much money reprogramming) • Positive: It will be a good building when finished - breaking the mind set of the initial proposal (addition to an existing building--bad idea)
FC (user)	• To achieve flexibility in the end use of the facility. We must look at what the next 50 years may have in store for us as users of the facility.	1. Function of the Laboratory space: - How will the space meet the needs of the program? - How do the spaces relate to one another? - Do they facilitate teaching? 2. Kitchen (function) 3. Computer lab (function)	Note: The user did not see a "program" • Negative: The biggest failure was that the initial program did not get updated to coincide with a major change in scope.

Table D.4: Construction Project C1--Critical Program Information

PROJECT	O. REP'S PRIMARY OBJECTIVE	CRITICAL PROGRAM ELEMENTS	LESSONS LEARNED
• C 1 (Research West) PM--	• Drawing out the user's needs, then transferring them to the AE	1. Schedule 2. Cost (it goes without saying that there wasn't enough money)	• <u>Negative:</u> - Time--the program and design development needed more control (i.e., setting up tighter, definite milestones, then meeting them) • <u>Positive:</u> - Overall, the equipment dat sheet was a good tool (should modify the process though--ie., provide better instruction, closer coordination)
PL/FC--	• Identify research committed by college to new faculty members	1. Time (the user needed it a year ago 2. Money (shortage of funds required down-scoping the project)	• <u>Negative:</u> - You can probably never be thorough enough (developing the program/reviewing documents...) • <u>Positive:</u> - looked at ways to cut corners to provide functions needed or save money.

Table D.5: Construction Project C2--Critical Program Information

PROJECT	O. REP'S PRIMARY OBJECTIVE	CRITICAL PROGRAM ELEMENTS	LESSONS LEARNED
• C 2 (Classrm Office Bldg) PM--	• Satisfy program requirements • Meet the given budget • "Fit the shoe to the foot" (achieve the needed function w/i budgetary constraints)	1. The design and construction to date have been good due to the "chemistry" between the designer and contractor (the author considers this answer a misinterpretation, but it was repeated, even when the question was rephrased)	• It is beneficial to "preprogram" (note- this refers to programming) or pre-plan. If this would have been done, we could have put the project to bed two years earlier. • The "pre-plan" should establish the budget, instead of the budget establishing the project scope.
PL--	• The first priority was the classroom spaces: - improve quality - release space elsewhere on campus • Release space for office functions	1. The function of the classroom "activities" 2. Keep the classrooms located on the first two floors. 3. The office spaces would follow suite (conform to classroom constraints)	• Do a program (there was no firm program here) • Positive: There is a value to student involvement (in planning committees)

Table D.3: Post-Occupancy Projects PO1 and PO2--Critical Program Information

PROJECT	O. REP'S PRIMARY OBJECTIVE	CRITICAL PROGRAM ELEMENTS	LESSONS LEARNED
• PO 1 (Hallowell Building) PL--	• Provide staging space during renovations affecting other departments (note: the building evolved as permanent space, rather than being utilized as originally planned)	1. Maximum utilization (for the user's functional requirements) based on maximum flexibility for the FC. 2. Cost 3. Time (they needed it for staging areas and other temp. space requirements)	• get more user input next time • there should be more than one person developing the program • additional rest rooms should nave been provided. (the AE didn't think so, and PL should have "stuck to their guns")
• PO 2 (Ruth bldg) PL--	• Time • Develop a flexible program (in case location for the occupant changed) • "SPACE"--went from 2300 to 1900 NASF • satisfy the "politics" behind having a new head of the research center	1. Educating users of the facility that it had relocated 2. Utilizing specific equipment (office high density storage) req. 3. Getting the most efficiency out of the space: - due to less SF - due to profitability y of leased space.	• Develop the program fully before "drawing lines on paper" • Institutionalize the system used for programming this project (a highly structure and thorough space planning study) • Budgeting and funding projects: - we (OPP) tend to approach funding unrealistically. - problems carrying money over from previous fiscal years